

Jani Huuhtanen

# Tasosuunnittelun menetelmät ja työkalut mobiilipeleissä

Metropolia Ammattikorkeakoulu

Insinööri (AMK)

Tieto- ja viestintätekniikan tutkinto-ohjelma

Insinöörityö

8.5.2018

Tekijä Otsikko	Jani Huuhtanen Tasosuunnittelun menetelmät ja työkalut mobiilipeleissä
Sivumäärä Aika	44 sivua + 1 liite 8.5.2018
Tutkinto	Insinööri (AMK)
Tutkinto-ohjelma	Tieto- ja viestintätekniikka
Ammatillinen pääaine	Pelisovellukset
Ohjaajat	Lehtori Antti Laiho Projektiasiantuntija Juha Huhtakallio
<p>Insinööritöiden tavoitteena oli luoda tasoja 2D-mobiilipeliin hyödyntäen alan ammattilaisten luomia menetelmiä ja tutustua tasosuunnittelussa käytettäviin työkaluihin sekä kehittää niitä. Työssä perehdyttiin pelin luomisen tuotantoprosessiin, ja siinä keskityttiin tarkastelemaan sen merkitystä erityisesti tasosuunnittelijan näkökulmasta. Työssä luodut tasot tehtiin käyttäen Unitya, joka on monialustainen pelimoottori.</p> <p>Pelinkehitys on monivaiheinen prosessi, joka vaatii usean osa-alueen osaamista. Tasosuunnittelu on tärkeässä roolissa jokaisen pelin kannalta, sillä se luo maailman ja olosuhteet, joissa pelaajat voivat toteuttaa itseään. Jotta suunnittelutyöstä ja toteutuksesta saadaan johdonmukainen, on tasosuunnittelijalla käytettäväänään paljon erilaisia suunnittelumenetelmiä ja työkaluja. Eri alustoille kohdistetuissa peleissä tulevat vastaan niiden luomat rajoitteet, joiden mukaan tasosuunnittelijan tulee mukautua ja pystyä luomaan toimiva rakenne pelattavaksi. Näistä riippumatta on eri menetelmiä ja työkaluja, joiden periaatteet ja toimintatavat pätevät kaikkeen tasosuunnitteluun.</p> <p>Työssä selvitettiin, miten eri menetelmien avulla suunniteltujen ja toteutettujen tasojen lopputulokset eroavat toisistaan. Tasosuunnitteluprosessille tehtiin yhteneväinen ja toimiva rakenne, jonka avulla pystytään jatkossa tuottamaan tasoja mobiilipeleihin.</p> <p>Insinööritöissä todettiin tärkeäksi pitkälle viedyn suunnitelman laatiminen ja sen noudattaminen, minkä avulla pystyttiin hallitsemaan isoakin kokonaisuutta. Työssä toteutettiin kolme tasoa, jotka siirtyivät jatkokehitykseen peliä varten. Tasot toteutettiin käyttäen työssä esitetyjä menetelmiä.</p>	
Avainsanat	tasosuunnittelu, mobiili, pelisuunnittelu, Unity, pelin kehitys

Author Title	Jani Huuhtanen Level design methods and tools in mobile games
Number of Pages Date	44 pages + 1 appendices 8 May 2018
Degree	Bachelor of Engineering
Degree Programme	Information and Communication Technology
Professional Major	Game Applications
Instructors	Antti Laiho, Senior Lecturer Juha Huhtakallio, Project Specialist
<p>The goal of this thesis was to create levels to a 2D mobile game, by using methods created by the industry professionals and to develop tools used in level design. This thesis goes through the production process of game development and focuses particularly on examining production process phases in level design. Levels created during this final year project was produced with Unity, which is a multiplatform game engine.</p> <p>Game development is a multiple phased process, which requires expertise from several fields. Level design has an important role, as it creates the world and circumstances where the players expresses themselves. For the design and development to be consistent, the level designer has an abundance of methods and tools to utilize.</p> <p>This final year project demonstrates the differences that occur when using different design methods. Level design process aimed to create a working and solid framework, that could produce levels for 2D mobile games in the future.</p> <p>As a result, it was found out that creating a proper design and applying to it, would allow to govern large projects. Three levels were created by applying these methods during this final year project.</p>	
Keywords	level design, mobile, game design, Unity, game development

# Sisällys

## Lyhenteet

1	Johdanto	1
2	Mobiilipelituotanto	2
2.1	Pelinkehityksen vaiheet	2
2.2	Tuotantoryhmän roolit ja toiminta	5
2.3	Tasosuunnittelun merkitys ja osuus pelinkehityksessä	9
3	Tasosuunnittelun prosessi ja pelimootorit	13
3.1	Menetelmät ja työkalut tasosuunnittelussa	13
3.2	Pelimootorit ja kenttäeditorit	18
4	Tasosuunnittelun toteutus ja työkalujen hallinnointi	21
4.1	Suunnittelutyön alustus	21
4.2	Tasorakenteen ja pohjapiirustusten luominen	23
4.3	Hahmotusmenetelmien hyödyntäminen ja prosessin muokkaaminen	25
4.4	Kentänluontimenetelmä 1: Kerrostus	29
4.5	Kentänluontimenetelmä 2: Ositus	31
4.6	Kentänluontimenetelmä 3: Valmiit palat	33
4.7	Tasosuunnittelutyökalujen käyttö ja räätälöinti	34
5	Tasosuunnitteluprosessin tarkastelu	36
5.1	Tulokset ja kokonaisuus	36
5.2	Esitettyjen menetelmien ja työkalujen hyödyntäminen jatkossa	39
6	Yhteenveto	40
	Lähteet	42

## Liitteet

Liite 1. Viimeistellyt tasot käytettyjen menetelmien mukaisessa järjestyksessä

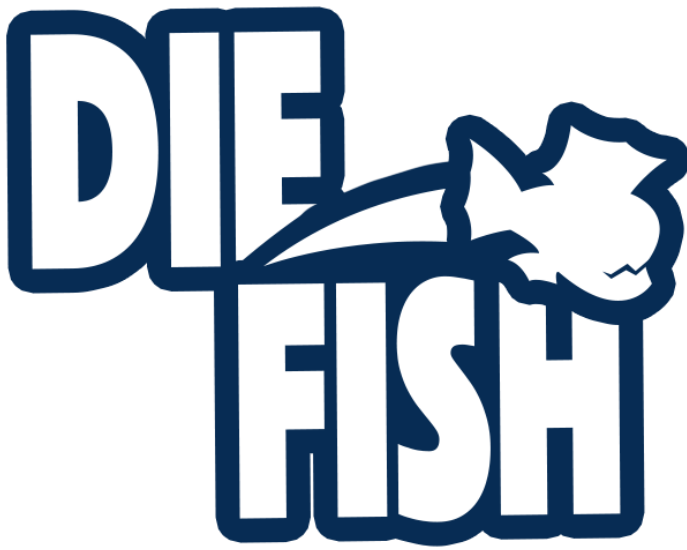
## Lyhenteet ja käsitteet

GDD	Game design document. Pelisuunnittelussa käytetty dokumentaatio, jonka pohjalta tuotantoryhmä toteuttaa peliä.
Mesh	Monikulmioista koostuva muoto, jolla määritetään graafisen elementin visuaalinen ilmentymä.
Export	Menetelmä, jolla saadaan luotua määriteltyä tiedostotyyppiä oleva tiedosto.
Level editor	Tasosuunnitteluun käytettävä ohjelma, joka sisältää kokoelman tarvittavia työkaluja.
Unity unit	Unityssa käytettävä yksikkö, jonka arvo on 1 Unity unit = 1 metri.
Place holder	Peliobjektin tilapäinen visuaalinen ilmentymä.

## 1 Johdanto

Insinööriyön tarkoituksena on tarkastella mobiilipelin tuotannon vaiheita ja keskittyä tasosuunnittelun merkitykseen 2D-peleissä. Insinööriyössä esitellään käytettäviä menetelmiä ja työkaluja. Työssä tutustutaan aiemmin kehitettyihin ja käytettyihin menetelmiin, joiden avulla prosessista saadaan toimiva ja johdonmukainen. Työssä myös kehitetään ja luodaan pelin muokkausohjelmassa käytettäviä työkaluja.

Insinööriyön osana toteutetaan tasoja Wave 7 Games -pelistudion Die Fish -mobiilipeiliin. Kuva 1 esittää pelistudion artistin tekemän logon Die Fish -peliin. Peliin luotavien tasojen toteutuksessa hyödynnetään eri menetelmiä ja työkaluja ja tarkastellaan niiden prosessiin tuomia eroavaisuuksia. Luodut tasot toteutetaan Unity-pelimoottorin kenttä-editorilla, ja työssä tarkastellaan ohjelmassa käytössä olevien ja luotujen työkalujen toimivuutta sekä haasteita.

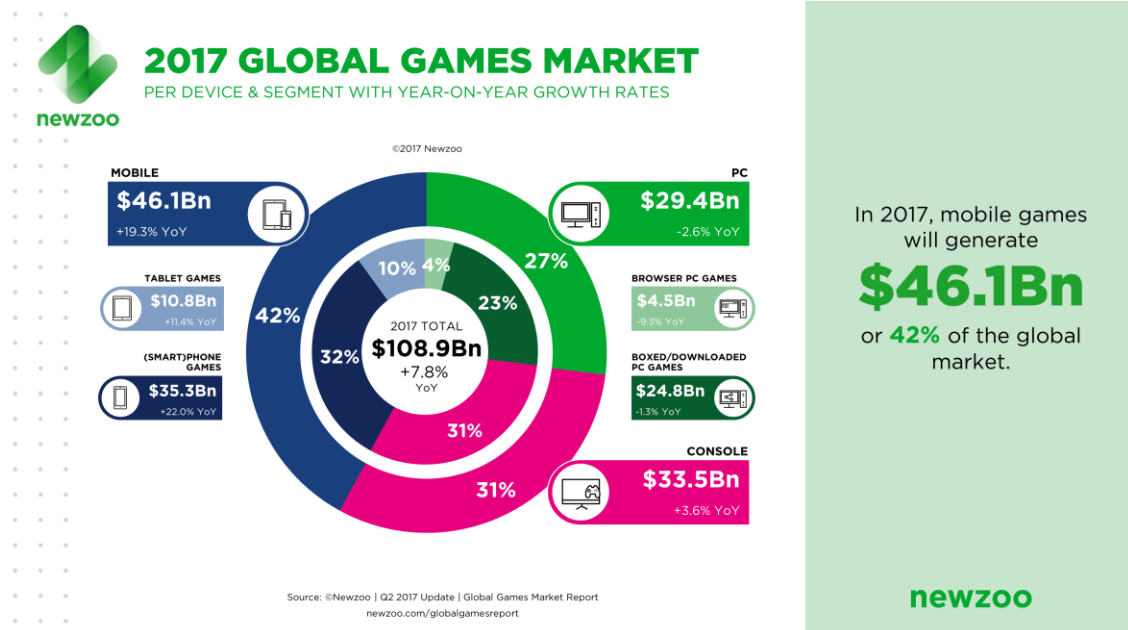


Kuva 1. Wave 7 Games -pelistudion Die Fish -logo.

Pelialan kehittyminen globaaleilla markkinoilla viimeisten vuosien aikana on ollut nopeaa. Suomalaiset pelialan yritykset ovat nousseet kansainvälisten markkinoiden mukana, ja peliala kattoi vuonna 2016 noin 0,5 % Suomen bruttokansantuotteesta. [1.]

Suomalaiset pelialan yritykset tunnetaan parhaiten mobiilipelien kehittämisestä, ja suurimmat mobiilipeleihin suuntautuneet yritykset tuottavat yli 90 % koko alan liikevaihdosta. Mobiilipelien osuus globaaleilla markkinoilla vuonna 2017 oli noin 42 % ja ne tuottivat

46,1 miljardia dollaria, kuten kuvasta 2 voidaan havaita. Tämä heijastuu myös kotimaisiin pelimarkkinoihin. [3; 2.]



Kuva 2. Globaalien pelimarkkinoiden liikevaihto vuonna 2017 ja sen jakauma laitteiden mukaan [2].

Pelin luominen voi tapahtua yksittäisen henkilön projektina, tai se voi olla suuren luokan tuotantoprosessi, johon kuuluu satoja henkilöitä. Pelinkehitykseen kuuluu useita tärkeitä osa-alueita, ja jokaisen peliin tuoma panos vaikuttaa lopputulokseen. Tasosuunnittelu on erittäin isossa roolissa, kun se luo pelaajalle maailman, jossa hän toimii ja toteuttaa itseään. Suunnittelussa työskennellään tiiviisti muiden osa-alueiden kanssa niiden määrittämien parametrien mukaisesti.

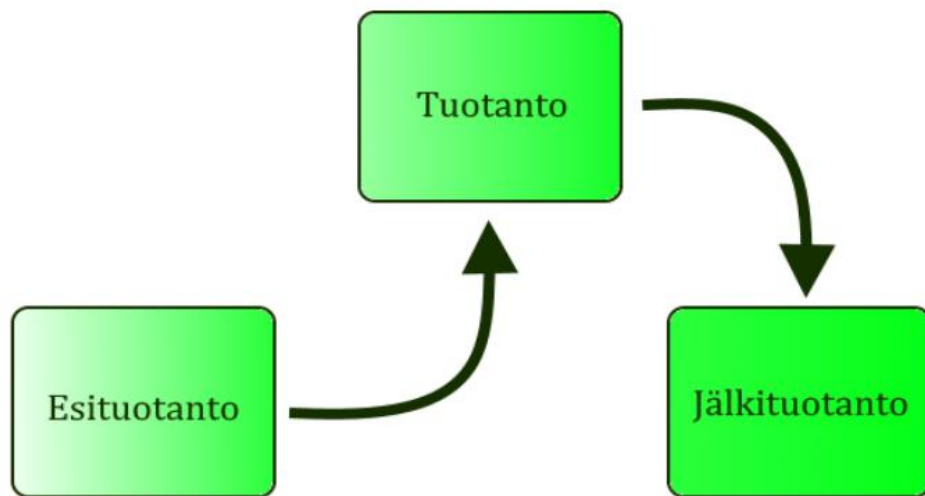
## 2 Mobiilipelituotanto

### 2.1 Pelinkehityksen vaiheet

Pelinkettäminen on monivaiheinen prosessi, joka käsittää kaiken konseptoinnista aina julkaisun jälkeisiin tapahtumiin. Jokainen vaihe pitää sisällään useita vaiheita, joiden avulla peliä työstetään julkaistavaksi tuotteeksi. Prosessin jakaminen omiin osa-alueisiin tuo työhön selkeyttä ja on tärkeä osa projektinhallinnan kannalta, jolloin aikataulujen luonti ja seuraaminen on helpompaa. [4.]

Kuten kuvassa 3 näkyy, pelinkehittäminen voidaan jakaa kolmeen vaiheeseen seuraavasti:

- esituotanto
- tuotanto
- jälkituotanto.



Kuva 3. Pelin kehittämisen vaiheet.

Jotta prosessissa edetään, täytyy jokainen osa-alue suorittaa huolellisesti, niin että tuotanto ei kärsi laadun tai aikataulun suhteen. Kuhunkin vaiheeseen tarvitaan usean osa-alueen osaamista, ja työskennellessä ryhmän kanssa on tärkeää kommunikoida jokaisen vaiheen etenemisestä.

Esituotanto alkaa pelin konseptoinnilla. Konseptoinnin tarkoituksena on luoda pohja, josta peliä ryhdytään toteuttamaan. Ryhmällä tai henkilöllä on ajatus pelistä, joka halutaan tehdä. Tätä ajatusta työestetään pidemmälle ja pyritään selvittämään pääpiirteisesti, minkälainen peli on kyseessä. [5.]



Konseptoinnin alussa idea voi olla hyvinkin yksinkertainen. Haastavampaa ideaa on tärkeä avata enemmän, jotta muut vaiheessa mukana olevat pystyvät osallistumaan kehittämiseen ja ymmärtävät, minkälaista peliä on tarkoitus lähteä luomaan.

Pelin konseptia työstettäessä voidaan miettiä vastauksia seuraaviin asioihin:

- genre
- alusta ja käyttöliittymä
- kohderyhmä
- tavoite
- uniikit kohdat
- päämekaniikat
- yleiset ominaisuudet
- tarinan yhteenveto
- monetisaatio
- referenssit ja konseptitaide.

Tarkempi konsepti auttaa seuraavan vaiheen työstämistä ja antaa mukana oleville henkilöille selkeämmän kuvan tulevasta. Konseptoinnin aikana artistit luovat konseptitaidetta, jolla pyritään hakemaan pelin tyyllistä suuntaa ja tuomaan ideoita pelin sisällön suhteen. [6, s. 3.]

Pelistä kirjoitetaan Game Design Document eli GDD, joka on pelin suunnitteludokumentti. Siihen sisällytetään kaikki pelistä tarvittava informaatio. Tätä dokumentaatiota käyttävät kaikki peliä työstävät henkilöt. Dokumentoinnilla pyritään luomaan pelille rakenne ja antamaan tuotantoryhmälle tarvittavat tiedot pelin toteuttamista varten. Tärkeää on ottaa huomioon eri osa-alueet, joiden tarvitsema informaatio eroaa osa-alueittain. Hyvä GDD on selkeä ja vastaa kaikkiin kysymyksiin, jotka liittyvät suunnitteluun ja pelin tapahtumiin. Dokumentaatiota päivitetään koko prosessin ajan, mutta alussa kehittäjille annetaan selkeä suunta, johon peliä viedään. [6; 12, s. 56.]

Esituotantovaiheessa ryhmä, joka koostuu tuottajista, suunnittelijoista, johtavasta ohjelmoijasta, johtavasta artistista ja käsikirjoittajista, tuottaa alustavaa materiaalia, jonka pelisuunnittelija kokoaa dokumentaatioon. Jokaisen jäsenen kontribuutio vaikuttaa dokumentin sisältöön. Dokumentissa pyritään kertomaan pelin tarina, ominaisuudet, mekaniikat, tasosuunnittelu, teknisen puolen tiedot, visuaalinen ulkoasu ja äänitehosteiden toiminta. [5.]

Pelin dokumentaatiota tehdessä on tärkeää pitää mielessä, minkälaista peliä ollaan luomassa ja mitä rajoitteita se tuo. Jokaisen tuotantoryhmän jäsenen pitää työstää osa-alueitaan samaan suuntaan, jotta tuotantoon siirryttäessä visio ja dokumentaatio ovat yhteneväiset.

Tuotantovaiheeseen siirryttäessä pelin työstäminen alkaa konkreettisesti. Tässä vaiheessa dokumentaatio on siinä vaiheessa, että muut tuotantoryhmän jäsenet pääsevät suorittamaan omaan toimenkuvaan kuuluvia tehtäviä, esituotantovaiheessa luodun aikataulun mukaisesti.

Jokainen ryhmä työskentelee omien osa-alueiden parissa, jotta peli saadaan julkaistavaksi. Ryhmien edistyessä ryhdytään yhdistämään eri alueiden komponentteja, jolloin varsinainen peli alkaa muodostua. Jokaisessa vaiheessa on tärkeää testata pelin toimivuutta ja muokata eri alueita, jotta saadaan aikaan mahdollisimman viihdyttävä ja toimiva lopputulos.

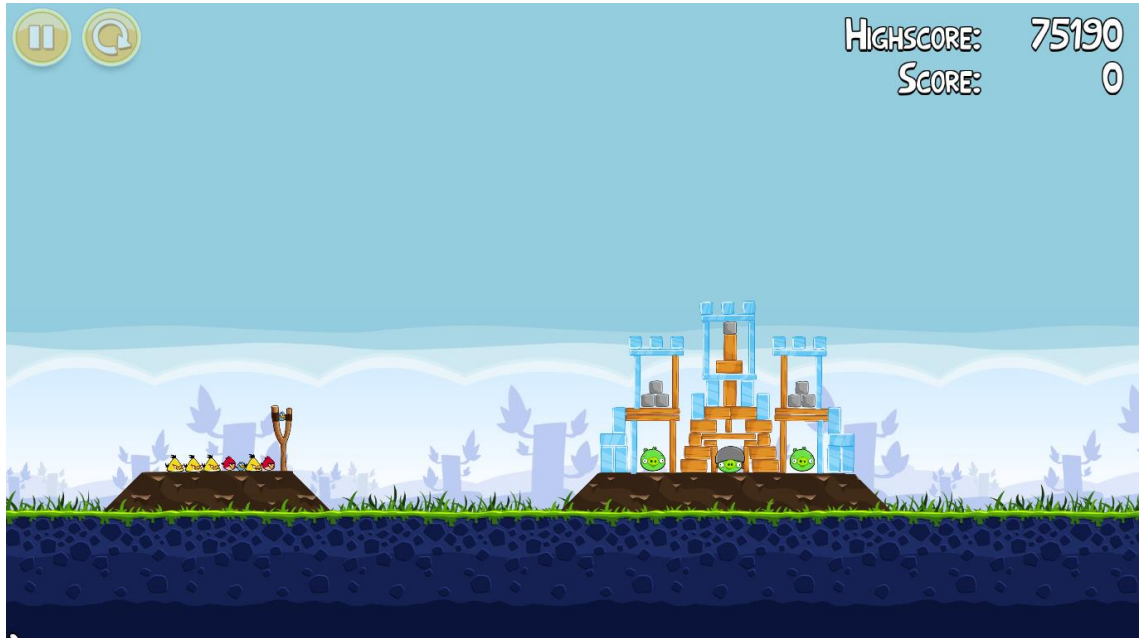
Pelinkkehittämisessä hyvin iteratiivinen vaihe on tuotannossa, jolloin pelistä luodaan eri koontiversioita testausta varten. Testauksen aikana pelistä käydään läpi eri toiminnallisuuksien ja mekaniikkojen toimivuus. Testauksessa tarkastellaan myös ääniefektien ja visuaalisten elementtien sopivuutta.

Peli siirtyy jälkituotantoon siinä vaiheessa, kun sen kaikki ominaisuudet on luotu ja koodaus ja taiteellinen puoli on valmis. Yleistynyt tapa on julkaista peli laajalle yleisölle jo beetavaiheessa, jolloin pelaamisesta saadun datan ja käyttäjien palautteen perusteella voidaan kehittää lopullista tuotetta. Pelistä julkaistaan Gold-versiona kutsuttu koontiversio, kun pelin katsotaan olevan valmis. Riippuen pelistä ja siihen liittyvistä rajoitteista, pyritään peliä markkinoimaan mahdollisimman varhaisessa vaiheessa, mutta viimeistään jälkituotantovaiheessa käynnistetään markkinointiprosessi. Monet pelit keräävät pelaajista ja pelistä pelaajien avulla dataa, joilla voidaan arvioida pelin toimivuutta ja näin ollen tehdä tarvittavia muutoksia pelikokemuksen parantamiseksi.

## 2.2 Tuotantoryhmän roolit ja toiminta

Pelinkkehityksessä mukana olevan tuotantoryhmän koko voi vaihdella yhden henkilön pienistä projekteista aina satoja henkilöitä työllistäviin julkaisuihin. Mobiilipeleissä

tuotantoryhmät ovat yleisesti ottaen pienempiä kuin suuriin pc-julkaisuihin osallistuvat. Esimerkiksi kuvassa 4 nähtävä Rovion Angry Birds Classic -pelin tekemisessä oli alun perin neljän hengen tuotantoryhmä. [9.]



Kuva 4. Rovion Angry Birds [24].

Kuvan 5 Blizzardin Overwatch-pelin tuotantoryhmän koko vaihteli, tuotannon eri vaiheissa neljästäkymmenestä aina sataan henkilöön. [8.]



Kuva 5. Blizzardin Overwatch [25].

Keskeisessä roolissa pelinkehityksen kannalta tuotantoryhmässä ovat

- tuottaja
- pelisuunnittelija
- ohjelmoija
- artisti
- animaattori
- äänisuunnittelija
- tasosuunnittelija.

Pienessä tuotantoryhmässä on yleistä, että yhdellä henkilöllä on useampi rooli. On kuitenkin hyvä muistaa, että jokaiseen rooliin pitää katsoa parhaiten soveltuva henkilö. Todennäköisesti artisti on pienessä ryhmässä paras henkilö tekemään myös animaatioita, mutta mikään ei estä ohjelmoijaa tekemästä myös sen osa-alueen töitä. Samoin esimerkiksi tasosuunnittelijasta voi tulla hyvä pelisuunnittelija, mutta on hyvin mahdollista, ettei hyvästä pelisuunnittelijasta tule hyvää tasosuunnittelijaa [12, s. 51].

Tuottaja valvoo eri ryhmien työskentelyä ja pitää huolen, että prosessi kulkee aikataulun mukaisesti. Tuottajan tehtävänä on myös valvoa, että pelin budjetti pysyy hallinnassa ja

pitää huolta pelin laadusta. Hän työskentelee suunnittelijoiden ja eri ryhmien vastuuhenkilöiden kanssa pysyäkseen ajan tasalla prosessin etenemisestä.

Pelisuunnittelija määrittelee pelin säännöt ja toiminnallisuuden: sen minkälaisia tapahtumia ja prosesseja pelaaja kohtaa pelissä. Hän luo suunnitteludokumentin, jonka avulla muu tuotantoryhmä työstää peliä. Pelisuunnittelijan tulee tietää yksityiskohtaisesti, kuinka eri osa-alueet pelissä toimivat. Suunnittelijan tulee huolehtia, että dokumentaatio on ajan tasalla. Pelisuunnittelijan tulee kommunikoida muiden tuotantoryhmien jäsenten kanssa, jotta luotu visio säilyy ja peliä työstetään tahdottuun suuntaan.

Ohjelmoija vastaa pelin teknisestä toteutuksesta. Hän työskentelee pelimoottorin parissa ja luo sillä rakenteen, jonka päälle peli tehdään. Ohjelmoija tekee peliin suurimman osan ominaisuuksista ja implementoi tarvittaessa muiden tuotantoryhmien jäsenten tuotokset. Ohjelmoija myös tuo oman lisänsä suunnitteluun, sillä toteuttaessaan eri ominaisuuksia hän luo toimivia ratkaisuja ominaisuuksiin.

Artisti tekee peliin kaiken tarvittavan grafiikan ja vastaa pelin visuaalisesta ja taiteellisesta puolesta. Hän luo konseptitaidetta, jonka pohjalta pystytään tarkentamaan pelin visuaalista tyyliä ja suunnittelemaan uusia elementtejä peliin. Pelin visuaalinen taide luo yleisesti pelaajan ensivaikutelman ja on kokonaisuuden kannalta hyvin tärkeässä roolissa, sillä ihmisen hallitsevin aisti on näkö. [10.]

Animaattori tekee pelin visuaalisiin elementteihin liikkeitä ja näin herättää pelin henkiin. Hän työskentelee tiiviissä yhteistyössä artistin kanssa, jotta jokaisen animoitavan elementin liikkeet ja ulkonäkö saadaan toimivaksi.

Peliin tulevien äänien, kuten muusikin ja ääniefektien, suunnitteleminen ja luominen kuuluu äänisuunnittelijalle. Äänien merkitys pelissä korostuu, kun efektejä käytetään pelaajalle annettavan informaation lähteenä. Peliin äänillä voidaan luoda tunnelma ja näin vahvistaa visuaalisten elementtien luomaa vaikutusta. Äänit viimeistelevät pelin kokonaisuuden ja luovat pelaajille vahvempia tunteita sekä muistoja pelistä. Mobiilipeleissä on kuitenkin huomioitava, että osa pelaajista ei pidä ääniä päällä pelatessaan. Puhelimien kaiuttimet eivät myöskään tällä hetkellä yleensä toista ääntä stereona, eikä voida pitää itsestään selvyytenä, että jokainen pelaaja käyttäisi kuulokkeita. Tämän vuoksi äänen osuus mobiilipeleissä ei ole aivan yhtä merkittävä, kuin pc-peleissä. [11; 12, s. 266.]

Tasosuunnittelijan tehtävänä on rakentaa pelin maailma. Hän luo yksityiskohtaisen ympäristön, jossa pelaajat toimivat. Tasot voivat vaihdella lineaarisesti etenevistä aina avoimiin maailmoihin, joissa pelaaja voi vapaasti kulkea omia teitään. Pelaajille on tärkeää luoda mielenkiintoinen ja sopivan haastava ympäristö, jossa he voivat toteuttaa itseään. Tasosuunnittelija vastaa tasojen tapahtumista ja tehtävistä. Tasosuunnittelu riippuu hyvin vahvasti pelin genrestä ja ulottuvuudesta. Tärkeimpänä työkaluna suunnittelija käyttää tasoeditoria, jolla hän pystyy luomaan peliin vaadittavat tasot. Käytetty tasoeditori voi olla joko tietylle pelille studion itsensä luoma tai yleisesti saatavilla oleva, jolla voidaan luoda tietyille pelityypeille vaadittavia tasoja.

Riippumatta tuotantoryhmän koosta on ensiarvoisen tärkeää, että kaikki pelin parissa työskentelevät tietävät minkälaista peliä ollaan tekemässä. Kun jokaisen osa-alueen elementtejä aletaan yhdistämään, pystytään näkemään, kuinka hyvin pelin alkuperäinen visio ja ryhmän kommunikaatio on toiminut. Pelin tekeminen on hyvin iteratiivinen prosessi ja on monesti todettu, että peli ei ole ikinä valmis vaan siitä löytyy aina jotain parannettavaa. Hyvällä yhteistyöllä, kommunikaatiolla ja testauksella pelistä saadaan työstettyä toimiva ja viihdyttävä kokonaisuus. [12, s. 60.]

### 2.3 Tasosuunnittelun merkitys ja osuus pelinkehityksessä

Pelin maailmat ja kentät ovat tasosuunnittelijan vastuulla. Tasosuunnittelija työskentelee pelisuunnittelijan luomien sääntöjen ja rajoitusten mukaan sekä käyttää työssään ohjelmoijan ja artistien luomia elementtejä. Kun pelaaja liikkuu maailmassa ja kohtaa esteitä, jotka pitää selvittää päästäkseen etenemään, on tasosuunnittelija luonut ne hänelle. Ilman tasoja ja maailmoja olisivat pelit tyhjiä kuoria ilman sisältöä, jossa pelaajalla ei olisi mitään tehtävää. Ed Byrne [13, s. 11] sanoo pelin tasojen toimivan tarinan kerronnassa kuin luvut kirjoissa: ne jakavat tarinan osioihin, jolloin voidaan luoda tarinan kaari.

Tasosuunnittelu on vaativa ja monimuotoinen prosessi, joka riippuu pelin genrestä, teemasta, ulottuvuudesta ja alustasta. Vaikka alustalla on merkitys ja se asettaa omia rajoituksia tasosuunnitteluun, pätevät samat periaatteet ja menetelmät pääsääntöisesti kaikkiin alustoihin. On kuitenkin huomattava, että työstettäessä lineaarisilla tasoilla varusteltua 2D-peliä mobiilialustalle tai ensimmäisen persoonan ampumapeliä 3D:nä tietokoneelle vaaditaan erilaisia työkaluja ja toteutustapoja. Monien eri genrejen, ulottuvuuksien ja laitteiden vaikutus tasosuunnitteluun on huomattava.

Tasosuunnittelussa on suuria kontrasteja ja eroavaisuuksia, kun verrataan esimerkiksi kuvassa 6 olevaa Seriouslyn Best Fiends -mobiilipeliä ja kuvan 7 PlayerUnknown's Battlegrounds Mobile -peliä. Best Fiends -pelissä pelaajan täytyy valita merkkejä luodakseen ruudukkoon yhtenäisiä ketjuja. Pelaajan pitää saada tietty määrä yhdisteitä luotua, jotta hän pääsee etenemään. Tässä tasojen suunnittelussa on luotu erimuotoisia alueita, johon merkit piirtyvät ja vaikutetaan mahdollisten merkkien määrään.



Kuva 6. Seriouslyn Best Fiends [23].

Vertailukohtana on Bluehole studiosin alun perin tekemä PlayerUnknown's Battlegrounds, joka on tietokoneelle ensin luotu, mutta sittemmin myös mobiilille julkaistu ensimmäisen persoonan ampumapeli. Kuvassa 7 nähdään, kuinka pelaajat liikkuvat PlayerUnknown's Battlegrounds Mobile -version avoimessa maailmassa. Pelin tavoite on pyrkiä olemaan viimeinen pelaaja tai ryhmä hengissä eliminoimalla vastustajat. Tasot koostuvat realismia tavoittelevista alueista, joita varten on luotu pohjapiirustukset ja



mallinnettu erilaisilla tasosuunnittelun työkaluilla maailma, jossa on rakennuksia, kasvustoa ja erilaisia maastotyyppejä. Pelit eroavat toisistaan suuresti, mutta molemmissa peleissä on tasosuunnittelijan täytynyt suunnitella ja luoda pelaajille maailma, jossa he voivat toimia.



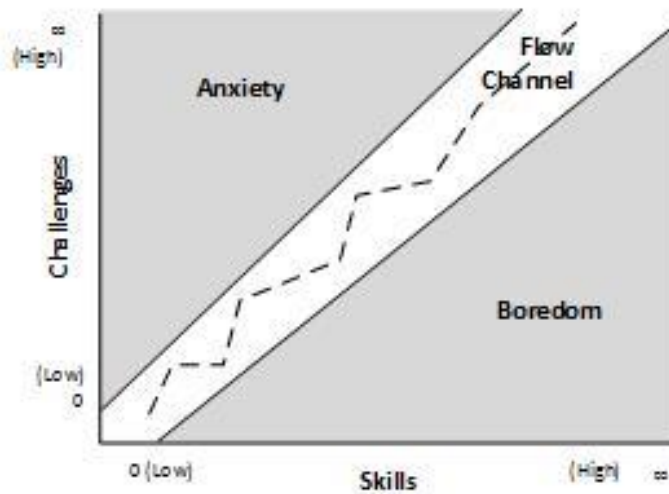
Kuva 7. PlayerUnknown's Battlegrounds Mobile -peli [26].

Lähtökohtaisesti pelin tasojen pitäisi olla pelaajalle viihdyttävä kokemus ja hyvin suunnitellut tasorakenteen tarjoavat pelaajille sopivilla intervaleilla kasvavia haasteita. Tasoja luotaessa suunnittelijan tulee miettiä, miksi hän haluaa luoda kyseisen tason, mikä tekee siitä hauskan ja pelattavan. Hauskuus on subjektiivinen käsite ja jokainen pelaaja kokee sen eri tavalla. Suunnittelijan on kuitenkin luotettava omaan intuitioon ja työstettävä peliin sopivia tasoja parhaan näkemyksensä mukaan. [22.]

Tasosuunnittelijan tulee ottaa huomioon, miten usein hän esittelee pelaajalle uusia elementtejä, jotta pelaajalla on aikaa oppia vaadittu asia. Mikäli pelaaja kokee, että hänen täytyy oppia mahdottoman paljon uusia asioita samanaikaisesti voi pelistä jäädä huomaamatta useita kohtia, joilla on merkitystä pelin suhteen jatkossa. Pelaaja voi lopettaa pelaamisen, mikäli oppimiskynnys on liian korkea. On myös mahdollista, että pelaajalle ei tarjota uusia haasteita tai elementtejä käytettäväksi tarpeeksi usein. Tällöin pelikokemus kärsii ja pelaaja voi kokea tasot liian helpoiksi, itseään toistaviksi ja tylsiksi. Suunnittelijan on tärkeä onnistua pitämään yllä pelissään sopivaa flow-tilaa ja tasojen vaikeus-tasojen, elementtien ja tapahtumien tulee tukea peliä kokonaisuutena. Flow-tilaa



kuvataan onnistuneeksi tasapainoksi pelin haasteiden ja pelaajan taitojen välillä sekä näiden kehitykseksi. Flow-tilaa visualisoiva kehitys voidaan nähdä kuvasta 8. [15.]



Kuva 8. Pelaajan kokeman flow'n kehitys haasteiden ja taitojen mukaan [15].

Mobiilipelien tasosuunnittelussa lisähaasteena ovat pelisessioiden kestot. Verrattuna keskimääräisiin konsoleilla ja tietokoneilla tapahtuviin pelisessioihin, ovat mobiilipelien keskiarvot huomattavasti pienemmät. Mobiilipelien sessioiden pituuteen vaikuttavat laitteen käyttötavat ja tapa, miten mobiilipelejä pelataan. Kun kuvassa 9 olevien tietojen perusteella lasketaan kaikkien pelaajien keskiarvo pelaajan ja pelisessioiden suhteen, käyttävät pelaajat noin 24 minuuttia päivässä pelaamiseen, ja jakavat sen neljään sessioon. Pelaajat kuitenkin jakautuvat useaan kategoriaan ja vähän pelaavien henkilöiden pelisessioiden keston keskiarvo laskee alle neljään minuuttiin ja heidän pelisessioidensa määrän keskiarvo alle yhteen päivässä. Tämän vuoksi on tärkeää osata suunnitella sopivanmittaisia tasoja peleihin, jolloin pelaaja pystyy etenemään pelissä yhden session aikana. [14.]

GAMER TYPE	PER DAY	PER MONTH	AVERAGE SESSION TIME
All Gamers:	4.31	133.6	5 min, 35 sec
Heavy Gamers:	10.6 (2.5x)	328.4 (2.5x)	5 min, 53 sec
Core Gamers:	16.6 (3.9x)	514.4 (3.9x)	6 min, 3 sec
Light Gamers:	0.6	17.1	3 min, 32 sec

Kuva 9. Eri pelaajatyypin pelisessioiden määrien ja kestojen keskiarvot mobiililaitteilla vuodelta 2016 [14].

Tasapainoiset ja sopivassa tahdissa etenevät tasorakenteet ja maailmat pitävät pelaajan mielenkiinnon yllä ja luovat parhaimmillaan kestäviä muistoja sekä saavat pelaajan palaamaan takaisin pelin pariin.

### 3 Tasosuunnittelun prosessi ja pelimoottorit

#### 3.1 Menetelmät ja työkalut tasosuunnittelussa

Jokaisella suunnittelijalla on itselleen toimivat menetelmät, mutta toimivien tasojen luomista varten on hyvä muistaa, että kunkin tason luominen alkaa huolellisella suunnitteluprosessilla. Suoraan tasonmuokkausohjelmaan meneminen ja tason luominen ilman käsitystä siitä, mitä ollaan tekemässä, mitä pelaajan halutaan kokevan tai kuinka pelaajan mielenkiinto pidetään yllä, johtaa todennäköisesti siihen, että taso jää kesken tai sitä ei tulla käyttämään valmiissa pelissä. Hyvin suunnitellut tasot pystyvät kertomaan tarinaa ja luovat pelaajalle tunteen etenemisestä, ja niillä on uudelleenpeluarvoa.

Vaikka tasosuunnittelu on pitkä prosessi usean huomioon otettavan seikan vuoksi, on yhtä tärkeää luoda visuaalisesti kiinnostava taso, jolla saadaan houkuteltua pelaajat pelaamaan, kuin tehdä tasosta hauska ja lisätä sinne tarpeeksi tehtäviä. Tasosuunnittelijalla on erilaisia menetelmiä varmistaa, että jokainen osa-alue otetaan huomioon jo suunnitteluvaiheesta lähtien. Ed Byrne [13, s. 56-58] listaa seuraavia kohtia, joita tasosuunnittelijan tulisi käydä läpi:

- Mikä tekee tasosta hauskan?
- Kuinka tason ergonomia toimii?

- Millainen on tason flow?
- Kuinka taso on rytmitetty?
- Miten haastava taso on?
- Mikä on tason innostava elementti?
- Millaisia koukkuja tasossa on?

Listan avulla voidaan miettiä suuriakin kokonaisuuksia ja luoda tasorakennetta, jonka pohjalta tehdään kaikki pelissä käytettävät tasot.

Tasosuunnittelijalla on myös useita eri välineitä, joita hän voi hyödyntää tehdäkseen omasta työstään järjestelmällisempää, nopeampaa ja tehokkaampaa. Vaikka varsinaisten tasojen luominen tehdään level editorissa eli kenttäeditorissa, kannattaa jokaisella suunnittelijalla olla seuraavia välineitä käytössään:

- kyniä ja vihko, johon voi kirjata ideoita ja ajatuksia
- taulu ja tusseja, johon voi luonnostella pohjapiirustuksia
- piirtotyökalu koneelle, jolla voi luoda viimeistellympiä ja helposti muokattavia ideoita tasoista.

Näiden välineiden avulla suunnittelija voi luoda pohjaa omille tasoilleen ja varmistaa, etteivät hyvät ideat unohdu. Kun muistaa dokumentoida ja tehdä alustustyötä huolellisesti on seuraavien vaiheiden aloittaminen ja tekeminen paljon miellyttävämpää.

Suunnittelun alussa on hyvä tutustua pelin teemaan tarkemmin ja hakea paljon lähdemateriaalia ideointia varten. Lähdemateriaalina voi käyttää esimerkiksi kirjallisuutta, elokuvia, luontoa, urbaaneja ympäristöjä, musiikkia tai taidetta. Esimerkiksi jos peli sijoittuu moderniin kaupunkiin, on hyvä tutustua nykyaikaiseen arkkitehtuuriin ja kaupunkien rakennustapoihin, minkä kautta saa käsityksen siitä, miten omia tasoja voi lähteä suunnittelemaan ja mitä sinne voisi luoda. On myös hyvä pelata muita pelejä ja katsoa niissä olevia ratkaisuja tasosuunnittelijan näkökulmasta. Peliä pelatessa kannattaa miettiä mikä tekee sen tasoista toimivia ja mikä herättää itsessä mielenkiintoa. Ideointivaiheessa on tärkeää dokumentoida kaikki ajatukset ja ideat. Dokumentointi voi tapahtua kirjoittamalla, piirtämällä, kuvaamalla tai vaikkapa nauhoittamalla puhettaan. [18, s. 9-12.]

Kun ideointivaiheessa on koottu tarpeeksi materiaalia, sitä voidaan ruveta yhdistelemään toimivaksi kokonaisuudeksi. Tähän voidaan käyttää soludiagrammia, jonka avulla luodaan sekvensseillä eräänlainen käsikirjoitus siitä, miten tasot etenevät ja mitä pelaaja

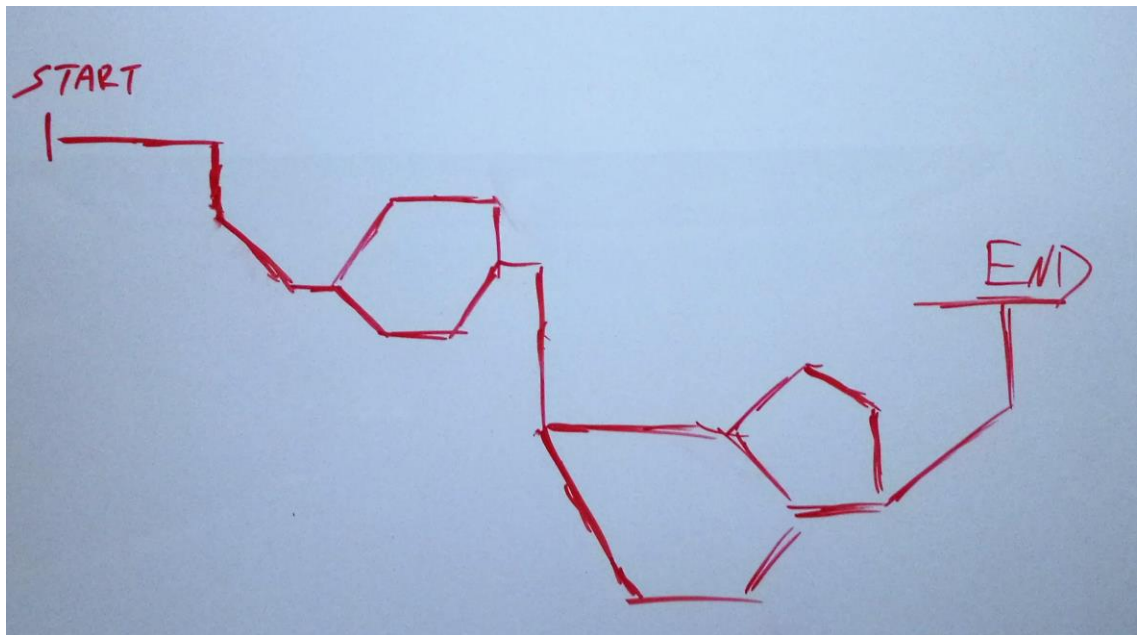
tulee kohtaamaan. Samassa vaiheessa on hyvä koota kaikki käytettävät elementit yhteen ja tehdä niille oma taulukko. Taulukon avulla on tarkoitus järjestellä elementit siihen järjestykseen, jossa ne esitellään pelaajalle: missä tasossa pelaaja kohtaa kyseisen elementin ensimmäisen kerran ja missä tasoissa kyseinen elementti on käytössä. [13, s. 148; 17.]

Kuvassa 10 nähdään, miten taulukon avulla voidaan esittää pelissä olevien elementtien esittely ja toistuminen tasoissa.

	Easy NPCs (weak and stupid)	Medium NPCs (medium strong, smart +/- fast)	Hard NPCs (fast, strong, smart and defense capable)	Grenadier NPCs	Sniper NPCs	Rocket Launcher NPCs	Ammo Caches	Health Packs	Projected Avg. Player Deaths
<b>Mission 1</b>	<b>0</b>						<b>5</b>	<b>8</b>	<b>0</b>
<b>Mission 2</b>	X						5	7	1
<b>Mission 3</b>	X						4	6	2
<b>Mission 4</b>	X	0		0			4	5	2
<b>Mission 5</b>	X	X			0		3	5	3
<b>Mission 6</b>	X	X			X	0	3	4	3
<b>Mission 7</b>	X	X	0	0		0	2	4	4
<b>Mission 8</b>	X	X	X	X	0		2	3	4
<b>Mission 9</b>	X	X	X		X	X	1	3	5
<b>Mission 10</b>	X	X	X	X		X	0	3	8
<b>0 = Introductory / Low Frequency    X = Part of regular repertoire of enemies</b>									

Kuva 10. Taulukko näyttää, miten eri elementit esitellään pelissä ja kuinka ne esiintyvät tasoissa. Kuvan taulukko on fiktiivisestä pelistä. [16.]

Seuraavassa vaiheessa tasoja aletaan luonnostella. Tasoista tehdään yksinkertainen pohjapiirustus, johon kuvataan, millainen taso on ja kuinka siinä edetään. Pohjapiirustus on hyvä tehdä helposti muokattavaksi, esimerkiksi piirtämällä lyijykynällä paperille tai tussilla taululle, kuten kuvassa 11, jolloin voidaan tehdä nopeita muokkauksia. Pohjapiirustuksia tehdään useita ja niitä tarkastellaan ja muokataan siihen asti, kunnes ollaan tyytyväisiä tason mittasuhteisiin ja aseteluun. [17.]



Kuva 11. Yksinkertainen pohjapiirustus 2D-peliin. Pohjapiirustus on tehty tussilla, jotta se on nopeasti muokattavissa.

Yksinkertaisten piirustusten jälkeen siirrytään luomaan yksityiskohtaisempia ja viimeistellympiä pohjapiirustuksia. Tähän käytetään hyväksi aiemmin luotuja piirroksia, joiden pohjalta työstetään tulevaa tasoa. Tässä vaiheessa tasoon aletaan lisätä pelillisiä elementtejä. Hyvä menetelmä on luoda symboleita pelissä käytettäville elementeille, tai jos käytettävissä on taidetta, voidaan se tulostaa ja leikata paloiksi. Näillä paloilla tai symboleilla merkitään tasoissa olevien elementtien paikka. Tämä helpottaa ja nopeuttaa tasojen muokkaamista ja antaa nopean tavan luoda eri variaatioita samoilla pohjapiirustuksilla. Tasoon pyritään lisäämään mahdollisimman paljon elementtejä, jotta tasomuokkastyökalua käytettäessä olisi hyvä malli, jonka pohjalta tasoa työstetään. [17.]

Tasosuunnittelijan tärkeimpiä työkaluja on kenttäeditori, jolla hän luo peliin suunnittelemansa tasot. Kun tason suunnitteluvaihe on suoritettu, siirrytään tasoa työstämään kenttäeditorilla aiemmin viimeistellyn pohjapiirustuksen mukaisesti. Tässä vaiheessa tason rakenne ja muoto hahmottuu paremmin. Vaikka pohjapiirustuksen olisi luonut kuinka huolellisesti, täytyy kaikki tason kohdat käydä läpi, sillä niitä joudutaan muokkaamaan kokonaisuuteen sopivaksi. Tämä johtuu osittain siitä, että pohjapiirustuksen piirtäminen täysin mittakaavaan on erittäin haastavaa ja jotkut osiot eivät toimi niin hyvin kuin alun perin on suunnitellut. Tason ei ole tarkoitus olla viimeistelty, vaan siihen lisätään geometriset elementit ja muita mahdollisesti tarvittavia elementtejä, kuten valaistus, jotta tasoa päästään pelaamaan ja nähdään, miten taso toimii kokonaisuutena. Tätä kutsutaan

myös whiteboxing-prosessiksi. Tässä vaiheessa käytetään yleisesti place holder -grafiikkaa eli tilapäistä peliobjektin visuaalista ilmentymää, jolloin tason luominen on nopeampaa eikä aikaa kulu grafiikan säätämiseen. [13, s. 210.]

Kaksi yleistä tapaa luoda tasoja on joko tehdä ne osissa tai niin sanotusti kerroksittain. Osien ja kokonaisten kenttien luonnissa järjestys ja toiminta on sama. Tasojen luominen kenttäeditorilla noudattaa yleisesti seuraavaa järjestystä:

- geometrinen ympäristö
- pelilliset elementit
- viimeistely koristeellisilla elementeillä.

Kun tasoja luodaan osissa, otetaan kokonainen taso ja pilkotaan se sopivankokoisiin osiin. Tällä tavoin kokonaisuus on helposti hallittavissa ja suunnittelija pääsee tekemään vaihtelevasti eri töitä eri osioiden parissa. Kun osat ovat sopivan pieniä, on helpompaa keskittyä yksityiskohtiin ja nähdä toimimattomat kohdat, joita tarvitsee muokata. Tämä luo kuitenkin myös omat haasteensa, sillä osia luotaessa voi puuttua vielä muun tuotantoryhmän materiaaleja, jolloin osioihin pitää palata myöhemmin tekemään muutoksia. Tämä lisää osiin käytettyä aikaa, ja suurien määrien kanssa on erittäin vaikea pitää huolta, että jokainen pala on päivitetty viimeisimmillä ja toimivilla elementeillä. [13, s. 223-225.]

Kokonaisten tasojen luonti kerroksittain tehdään muutoin samoilla periaatteilla kuin osissa luodut, mutta siinä kokonaisen tason yksi vaihe suoritetaan loppuun, ennen kuin siirrytään eteenpäin. Tämä tarkoittaa, että hallittavat kokonaisuudet ovat suurempia ja vaativat työstövaiheessa enemmän aikaa tasosuunnittelijalta. Tällä tavoin luotua kenttää pääsee testaamaan kokonaisuudessaan, aiemmin, ja se antaa muille tasoihin liittyville ryhmille tai henkilöille aikaa viimeistellä omia osioita ja nähdä miten niitä käytetään tasossa. Kerrostusmenetelmää käyttäen tasojen luonti on kuitenkin monotonisempaa. Kun tasoja ei nähdä viimeisteltynä eikä päästä työstämään eri vaiheita, saattaa toimimattomien ja virheellisten kohtien huomaaminen viivästyä ja näin haitata projektin aikataulua. Aikataulullisesti on myös mahdollista, että peli pitäisi saada julkaistua aiemmin, mutta valmiina on vain viimeistelemättömiä tasoja. [13, s. 225-227.]

Tasoja varten voidaan myös tehdä yksittäisiä kohtauksia tai paloja. Jos tasosuunnittelijalla on mielessään toimiva tai hauska kohta peliin, se kannattaa piirtää tai muulla tavoin luoda myöhempää käyttöä varten. Yksittäisten tasojen luomisessa voidaan

myöhemmässä vaiheessa käyttää aiemmin kuvattua menetelmää, jolloin kokonainen taso luodaan alusta ja siihen tulevat elementit asetellaan jokaiseen erikseen tai se voidaan tehdä hyödyntämällä kokonaan tai osittain valmiiksi koottuja paloja.

Tasoja täytyy testata ja pelata mahdollisimman paljon ja usein. Kun ensimmäinen vedos editorilla luodusta tasosta on valmis, käydään jokainen kohta tarkasti läpi ja katsotaan, onko taso oikeassa mittakaavassa. Suunnittelijan tulee muokata tasoa, jotta siitä tulee pelaajalle hauska ja toimiva. Todennäköisesti tasossa joudutaan muokkaamaan alustoja, joilla pelaaja liikkuu, sillä ne voivat olla liian helppoja tai haastavia, lisäämään tai poistaa tasossa olevia vastustajia sekä tasapainottamaan tasosta löytyvien elementtien määrää. Pelin mekaniikkojen ja fysiikoiden tulisi olla valmiit, kun tasoja aletaan luoda, sillä jokainen muutos voi rikkoa tehdyt tasot. Esimerkiksi jos pelaajan nopeutta hidastetaan ja hänen tulisi hypätä kuilujen yli, mutta nopeuden takia sitä ei voi enää tehdä, tällöin voidaan joutua muokkaamaan useita kohtia eri tasoissa, mikä taas vaikuttaa muihin elementteihin. [13, s. 233-234; 17.]

Kun tason rakenteeseen ja toiminnallisuuksiin ollaan tyytyväisiä, voidaan tasoa ryhtyä viimeistelemään. Tällöin kaikki tason elementtien grafiikka päivitetään ja lisätään äänitehosteet. Tasoa pitää testata ja pelata myös viimeistelyvaiheessa, sillä grafiikka vaikuttaa paljon tason tunnelmaan ja pelikokemukseen. Voi olla, että jotkut kohtaukset eivät toimi, koska grafiikka ei soinnu yhteen tai tekee siitä haastavan ymmärtää. Tässä tapauksessa pitää miettiä, pitääkö kohtauksessa vaihtaa elementtejä vai täytyykö grafiikkaa muokata sopivammaksi. Tämä riippuu vahvasti siitä, kumpi on nopeampaa ja miten ne vaikuttavat muihin tasoihin. Kun kaikki nämä kohdat on ratkaistu, pitäisi tason olla valmis pelaajille pelattavaksi [17].

### 3.2 Pelimoottorit ja kenttäeditorit

Pelimoottorit ovat jokaisen pelin ydin. Niiden avulla kehitetään ja luodaan erilaisia pelejä, ja ne kokoavat yhteen toiminnallisuuden, fysiikat, grafiikan, animaatiot, äänet sekä kaikki muut toimivan pelin aikaansaamiseksi vaadittavat elementit. Pelimoottoreita on useita erilaisia ja monet kehittävät omansa, jotta ne tehokkaasti vastaavat oman pelin tarpeita.

Pelimoottorit ovat tapa luoda yhtenevä runko, jonka avulla pelinkehityksessä jää enemmän aikaa yksilöllisten elementtien luomiseen. Viimeisten vuosien aikana on julkaistu

useita pelimoottoreita, joilla on saatu iso joukko uusia pelintekijöitä aina harrastelijoista ammattilaisiin. Useita tunnettuja pelimoottoreita ovat

- Unity 3D, <https://unity3d.com/>
- Unreal Engine, <https://www.unrealengine.com/>
- Cry Engine, <https://www.cryengine.com/>
- Source Engine
- GameMaker, <https://www.yoyogames.com/gamemaker>
- GoDot Engine, <https://godotengine.org/>
- Lumberyard, <https://aws.amazon.com/lumberyard/>.

Monet pelimoottorit tarjoavat käyttäjilleen visuaalisen käyttöjärjestelmän, jossa on käytettävissä eri kehitystyökaluja. Näillä työkaluilla käyttäjä pystyy luomaan hyvinkin nopeasti toimivia kokonaisuuksia ja moottorin avulla saadaan luotua varsinainen ohjelma, jonka kautta peli ajetaan. Pelimoottorien komponentteihin kuuluu yleisesti renderöinnin suorittava moottori, joka luo graafiset elementit jollain määrättyllä tekniikalla. Renderöintimoottori tarjoaa rajapinnan, jonka toteutuksen avulla hyödynnetään näytönohjainta. Äänimoottorin tehtävä on hoitaa pelissä toimivien äänien suorittaminen ja abstrahoida laitteen kanssa tapahtuva toiminallisuus käyttäjälle. Fysiikkamoottori mallintaa ja laskee kaikki pelissä tapahtuvat fysikaaliset ilmiöt.

Useissa pelimoottoreissa, kuten kuvan 12 Unreal Engineessä, on mahdollisuus luoda suoraan tasoja käyttämällä moottorin universaalia kenttäeditoria, jolloin ei tarvita erillistä kenttäeditoria. Näin monet tasot ja niiden elementit ovat käytettävissä useille peleille, ja niitä voidaan jakaa toisten kehittäjien kanssa. Yhtenä mallina voidaan tarkastella Unityn Asset Store –kauppaa, johon käyttäjät voivat lisätä tekemiään paketteja, jotka sisältävät tasoja varten tehtyjä elementtejä. Toiset käyttäjät voivat ladata ilmaiseksi tai ostaa paketin ja hyödyntää sitä omassa pelissään. [19.]

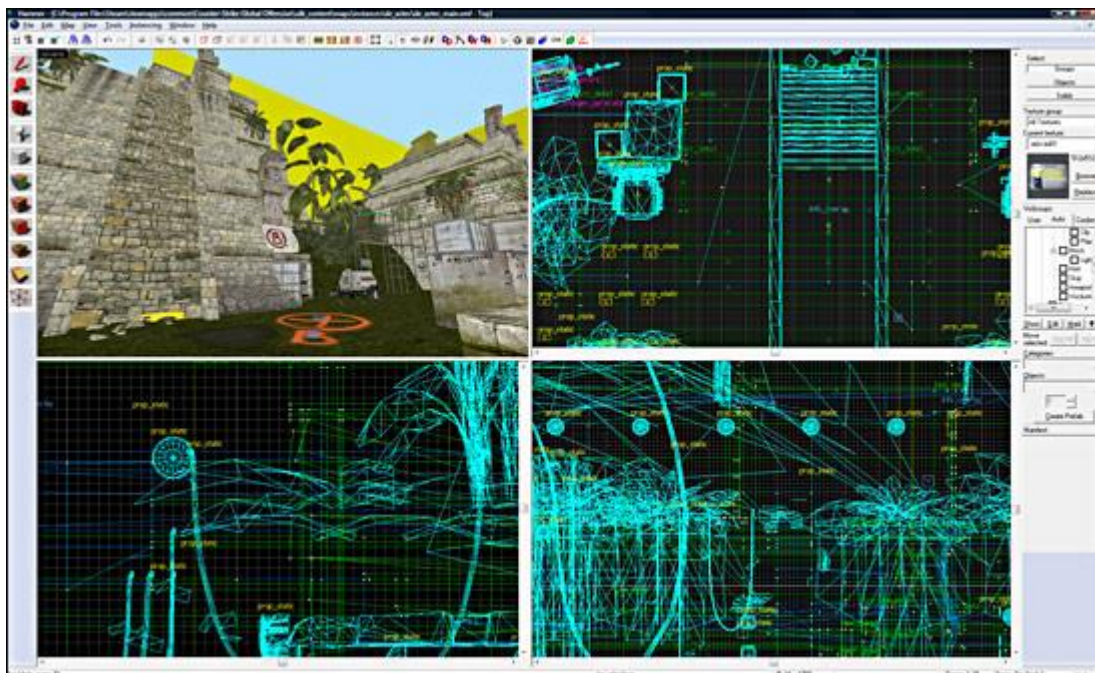




Kuva 12. Unreal Enginen kenttäeditori [20].

Käytettäessä universaalia editoria on käytettävissä suuri määrä erilaisia työkaluja, mutta käyttäjän täytyy huomioida, että niiden on tarkoitus soveltua moniin eri peleihin ja toimia eri ulottuvuuksissa. Monissa pelimoottoreissa voidaan luoda omia työkaluja ja muokata käytössä olevia, jolloin niistä saadaan omaan peliin paremmin sopivat.

Toinen mahdollinen tapa luoda tasoja on tehdä erillisellä ja juuri sitä varten ohjelmoidulla kenttäeditorilla, kuten kuvassa 13 olevalla Valven Hammer Editorilla. Tällaisen editorin suurimpia etuja on, että mikäli se luodaan itse, siitä voidaan tehdä juuri sellainen, kuin tasosuunnittelija haluaa. Kun kenttäeditoria luodaan, on hyvä tehdä lista, minkälaisia toiminnallisuuksia siihen haluaa. Ne käydään yhdessä ohjelmoijan kanssa läpi ja hän pyrkii toteuttamaan editorin vaadittujen parametrien mukaisesti. Erillinen kenttäeditori on myös usein kevyempi kuin pelimoottorin yhteydessä käytettävä, mutta ei tällöin sisällä mahdollisuutta pelata ja testata peliä, vaan tasolle tehdään export eli ohjelma luo tason pelimoottorin vaatimalla tiedostoformaattilla, jotta se voidaan lukea ja suorittaa pelissä.



Kuva 13. Valven Hammer-kenttäeditori, joka on suunniteltu Source-enginelle [27].

Pelimoottorin ja kenttäeditorin olisi oltava yhteneviä, eli tiettyä editoria käytetään sille tarkoitetun moottorin kanssa. On hyvin haastavaa luoda tasoja eri työkaluilla, kuin mitä peli käyttää, mutta sekin on mahdollista. Erillisen kenttäeditorin luonti on kuitenkin myös aikaa vievä prosessi, ja sitä luotaessa kannattaa miettiä, tuleeko sitä käyttämään jatkossa tai voiko sen osia hyödyntää muissa projekteissa.

## 4 Tasosuunnittelun toteutus ja työkalujen hallinnointi

### 4.1 Suunnittelutyön alustus

Insinööriyön osana tehtiin Wave 7 Games -pelistudion Die Fish -peliin tasoja hyödyntäen eri suunnittelumenetelmiä. Työssä tarkasteltiin tasojen suunnitteluprosessin kulkua ja tasojen luomista erilaisia menetelmiä ja työkaluja hyödyntäen. Die Fish on 2D:nä toteutettu parven hallintaan keskittyvä peli, jossa käytetään elementtejä platformer ja survival tyyllisuunnan peleistä. Peli suunniteltiin ja toteutetaan mobiilialustalle. Kuvassa 14 nähdään, miten pelaaja ohjaa mutatoituneiden piraijoiden parvea. Piraijoilla on tarkoitus selvittää reitti erilaisten haastavien tasojen läpi syöden vastaan tulevia hahmoja sekä pysyä elossa mahdollisimman suurella parvella. Die Fish sijoittuu vedenalaiseen maailmaan, jossa piraijat kohtaavat niin vaarattomia snorklaajia ja pikku kaloja kuin polttavia

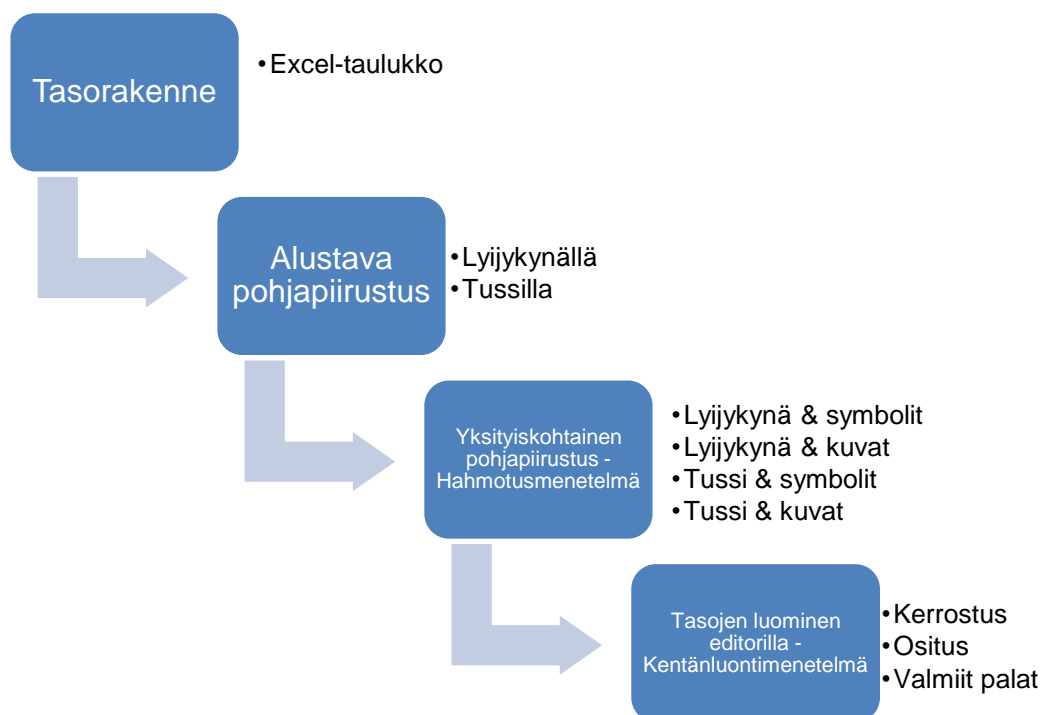
meduusoja ja harppuunakalastajia. Pelin yleisilme on rauhallinen, ja siinä käytetään hilitysti korostavia värejä luomaan elementeistä hyvin erottuvia ja mielenkiintoisia pelaajan silmissä. Pelimoottorina Die Fish käyttää Unitya. Unity tarjoaa oman integroidun kenttä-editorin, jossa on suuri määrä toteutuksessa vaadittavia työkaluja. Tämän lisäksi editoriin luotiin omia pelille spesifejä työkaluja.



Kuva 14. Wave 7 Games -pelistudion Die Fish -mobiilipeli.

Pelissä tulee valmiina olemaan kahdeksankymmentä erilaista tasoa. Insinööriyön aikana luotiin kolme eri tasoa aina suunnitteluvaiheesta lopulliseen viimeistelyyn versioon saakka. Tasoista ensimmäinen luotiin luvussa 2 kuvatulla kerrostusmenetelmällä, kun taas toiseen tasoon kokeiltiin osittamista ja kolmas taso luotiin käyttäen yksittäisiä paloja. Tasot sijoittuvat pelin jälkimmäiselle puoliskolle ja sisältävät näin suuremman määrän elementtejä sekä ovat pelaajalle vaativampia. Tasojen luonnissa kiinnitettiin erityistä huomiota siihen, miten tasot erosivat työmäärän ja keston suhteen, kun vaihdetaan toteutusmenetelmää. Peliin oli aiemmin työstetty tasoja erilaisilla menetelmillä ja poiketen täysin uudesta projektista oli työvaiheessa tarjolla suuri määrä pitkälle työstettyjä elementtejä. Näin ollen tietyt kohdat suunnittelussa poikkesivat esimerkiksi visuaalisten ulkoasujen osalta eikä luonnin aikana käytetty paljoa yksinkertaista place holder -grafiikkaa.

Kuvan 15 avulla nähdään, minkälainen tasojen luomisprosessi on ja kuinka siinä edettiin.



Kuva 15. Tasojen luomisprosessin kuvaus vaiheittain.

#### 4.2 Tasorakenteen ja pohjapiirustusten luominen

Ennen tasojen luomista kerättiin taustamateriaalia ja dokumentoitiin niiden kautta saatuja ideoita. Koska peli sijoittuu vedenalaiseseen maailmaan, oli hyvä tutustua erilaisiin merissä ja vesistöissä eläviin organismeihin sekä selvittää, miten erilaiset maastotyyppit muodostuvat merten pohjissa. Innoittajana tasoja luotaessa toimi BBC:n vuonna 2001 julkaistu luontodokumenttisarja *The Blue Planet*, jossa tutustaan kahdeksassa tunnin mittaisessa jaksossa maailman meriin. Koska peliin oli luotu jo useita elementtejä mekaniikkoineen, oli hyvä tarkastella niiden toimintaa myös sen mukaan, miten ne käyttäytyvät luonnollisessa elinympäristössään.

Tasoihin kerättyjen ideoiden ja inspiraation jälkeen piti kerätä kaikki tasoissa käytettävät elementit yhteen listaan ja pyrkiä luomaan toimiva rakenne, jonka pohjalta voitiin työstää tasoja tietyille kohdille tasorakennetta. Vaikeusastetta pyrittiin kasvattamaan tiettyjen tasojen välein, jolloin pelaaja pystyy kehittymään, mutta ei kuitenkaan tunne olevansa jatkuvan paineen alla, ja etsimään sopivat välit, milloin pelaajalle voidaan esitellä uusia

elementtejä, jotta hänen mielenkiintonsa pysyy yllä eikä hän joudu kerralla käsittelemään liikaa tietoa. Kuvasta 16 voidaan nähdä, kuinka rakenne tehtiin luomalla Excel-taulukko, johon merkittiin elementit ja tasojen määrä. Näin voitiin merkitä kullekin elementille paikka, jossa se esitellään ja se, missä tasoissa kyseisiä elementtejä tulee pelaajalle vastaan sen jälkeen.

Level progression table										
Level:	Basic fish	Sea urchin	Diver	Chemical	Jelly fish	Sea mine	Basic gate	Explosives	Spurter	Introduce
1										
2	x									
3	x									
4	x									
5	x	x								
6	x	x								
7	x	x								
8	x	x								
9	x	x								
10	x	x								
11	x	x								
12	x	x								
13	x	x	x							
14	x	x	x							
15	x	x	x							
16	x	x	x							
17	x	x								
18	x	x								
19	x	x								
20	x	x								
21	x	x								
22	x	x								
23	x	x								
24	x	x								

Kuva 16. Osa ensimmäisestä tasorakenteen luomista varten suunnitellusta kaaviosta.

Tasorakenteen kanssa haastavaa oli löytää siihen tasapaino, joka kestäisi koko pelin ajan ja olisi riittävä monelle eri pelaajaryhmälle. Rakenne muuttui useaan otteeseen ja on monilta osin vielä testattavana.

Kun tasorakenne oli saatu työstettyä tyydyttävään pisteeseen, josta voitiin lähteä rakentamaan tiettyyn vaikeustasoon soveltuvia kenttiä, päästiin aloittamaan alustavien hahmotelmien tekoa. Tässä vaiheessa kokeiltiin eri vaihtoehtoja hahmotelmien tekemiseen. Perinteisellä lyijykynällä paperille tehdyt toimivat hyvin, ja niiden etuina oli, että niitä voidaan luoda useita kappaleita, jotka voidaan arkistoida talteen ja palata työstämään niitä myöhemmin. Haastavaksi muodostui tason mahdollistaminen suuntaa antavassa mitta-kaavassa, mutta usean tehdyn hahmotelman jälkeen tämä alkoi luonnistua suhteellisen hyvin. Osa luotiin tussilla ja taululla. Tämä vaihtoehto oli nopeampi ja huomattavasti helpompi muokata. Haasteena oli kuitenkin jatkokäsittely, sillä jokainen hahmotelma piti erikseen kuvata talteen, ja jos niiden suhteen aikoi jälkikäteen tehdä muutoksia, ne piti piirtää uudelleen. Samalla kuvaaminen kuitenkin helpotti arkistointia ja hahmotelmien jakamista.

Alustavien hahmotelmien luontia varten ennakkoon tehdyn taustatutkimuksen ja materiaalien avulla saatiin nopeasti työstettyä kaksikymmentä erilaista pohjaa, joista voitiin valita parhaiten soveltuvat seuraavaan vaiheeseen jatkoon. Jatkokäsittelyyn päätyneiden hahmotelmien määrä oli lopulta viisi. Tämä on myös yleinen suhdeluku siitä, kuinka moni tehty kenttä päättyy peliin. Hahmotelmien valinnassa kiinnitettiin huomiota reittivalintojen toimivuuteen ja tason pituuteen.

#### 4.3 Hahmotusmenetelmien hyödyntäminen ja prosessin muokkaaminen

Jatkokäsittelyyn siirrettyjen hahmotelmien myötä edettiin suunnitteluprosessin seuraavan vaiheeseen. Tässä kohtaa päästiin hyödyntämään useaa eri menetelmää, jolloin prosessissa alkoi muodostua eroavaisuuksia. Yksityiskohtaisten ja viimeistelyjen pohjapiirustuksien luontiin käytettiin hyväksi tasorakennetta ja alustavia hahmotelmia. Samalla kokeiltiin, kuinka yksittäisten kohtausten suunnittelu toimii kokonaisten tasojen työstämisessä. Sekä tasojen yksityiskohtaisten pohjapiirustusten että palojen luonnissa kokeiltiin taulukossa 1 nähtävien menetelmien mukaan, kuinka eri symboleiden merkintä ja esitystapa vaikuttivat pohjapiirustuksen selkeyteen ja luettavuuteen.

Taulukko 1. Tasojen piirtämiseen ja dokumentointiin käytetyt hahmotusmenetelmät.

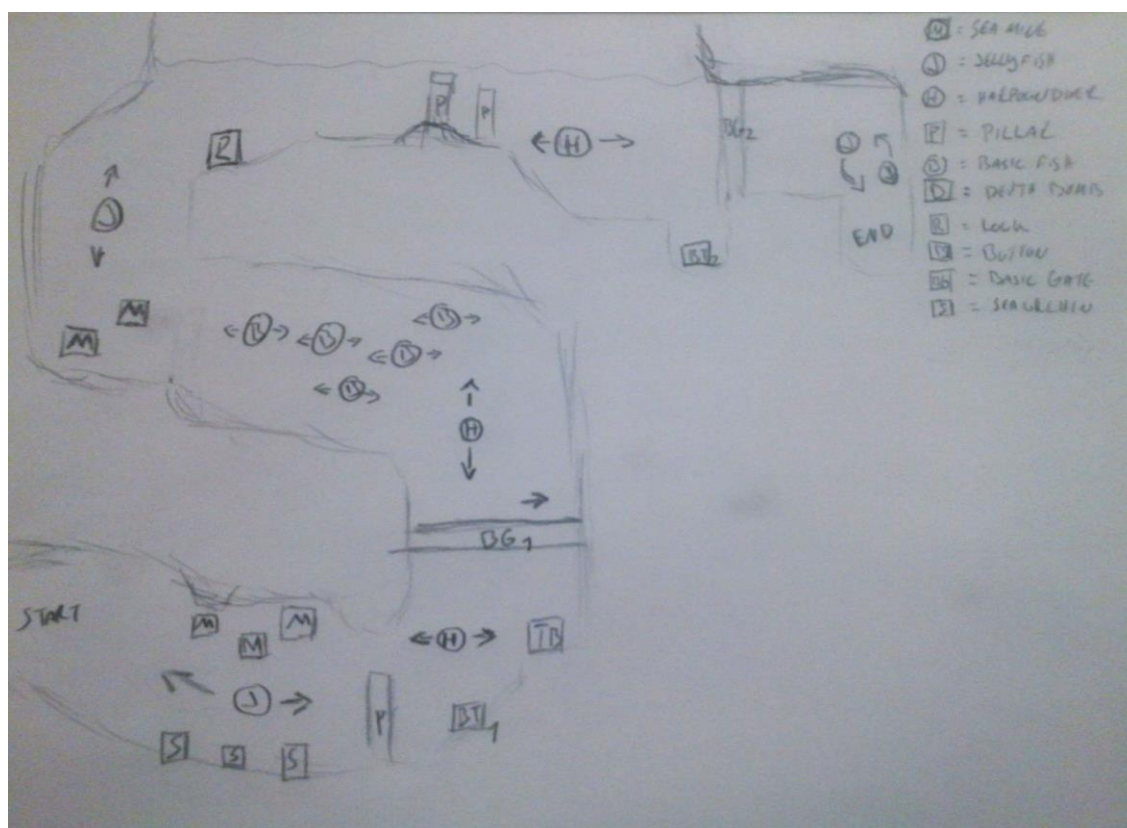
<b>Hahmotusmenetelmä 1:</b>		<b>Hahmotusmenetelmä 2:</b>
<b>Lyijykynä &amp; symbolit</b>		<b>Lyijykynä &amp; kuvat</b>
Piirros paperille		Piirros paperille
Elementtien lisäys symboleilla		Elementtien lisäys kuvilla
Arkistointi		Kuvaus
		Arkistointi
<b>Hahmotusmenetelmä 3:</b>		<b>Hahmotusmenetelmä 4:</b>
<b>Tussi &amp; symbolit</b>		<b>Tussi &amp; kuvat</b>
Piirros taululle		Piirros taululle
Elementtien lisäys symboleilla		Elementtien lisäys
Kuvaus		Kuvaus
Arkistointi		Arkistointi

Kustakin viidestä hahmotelmasta piirrettiin yksityiskohtaiset pohjapiirustukset, sekä paperille että taululle. Kaksi tasoista tehtiin ensin hyödyntäen lyijykynää ja paperia, minkä jälkeen sama taso luotiin taululle tussin avulla. Loput kolme tasoa aloitettiin taululla ja tehtiin tämän jälkeen samanlainen vedos paperille. Näin pystyttiin vertaamaan



lopputuloksia keskenään paremmin, ja molemmilla tavoilla jouduttiin luomaan yksityiskohtaisempi pohjapiirustus ensimmäistä kertaa. Elementtien lisäämistä varten määriteltiin jokaiselle käytettävälle elementille oma symboli ja käytettiin tulostettuja sekä leikat-  
tuja kuvia.

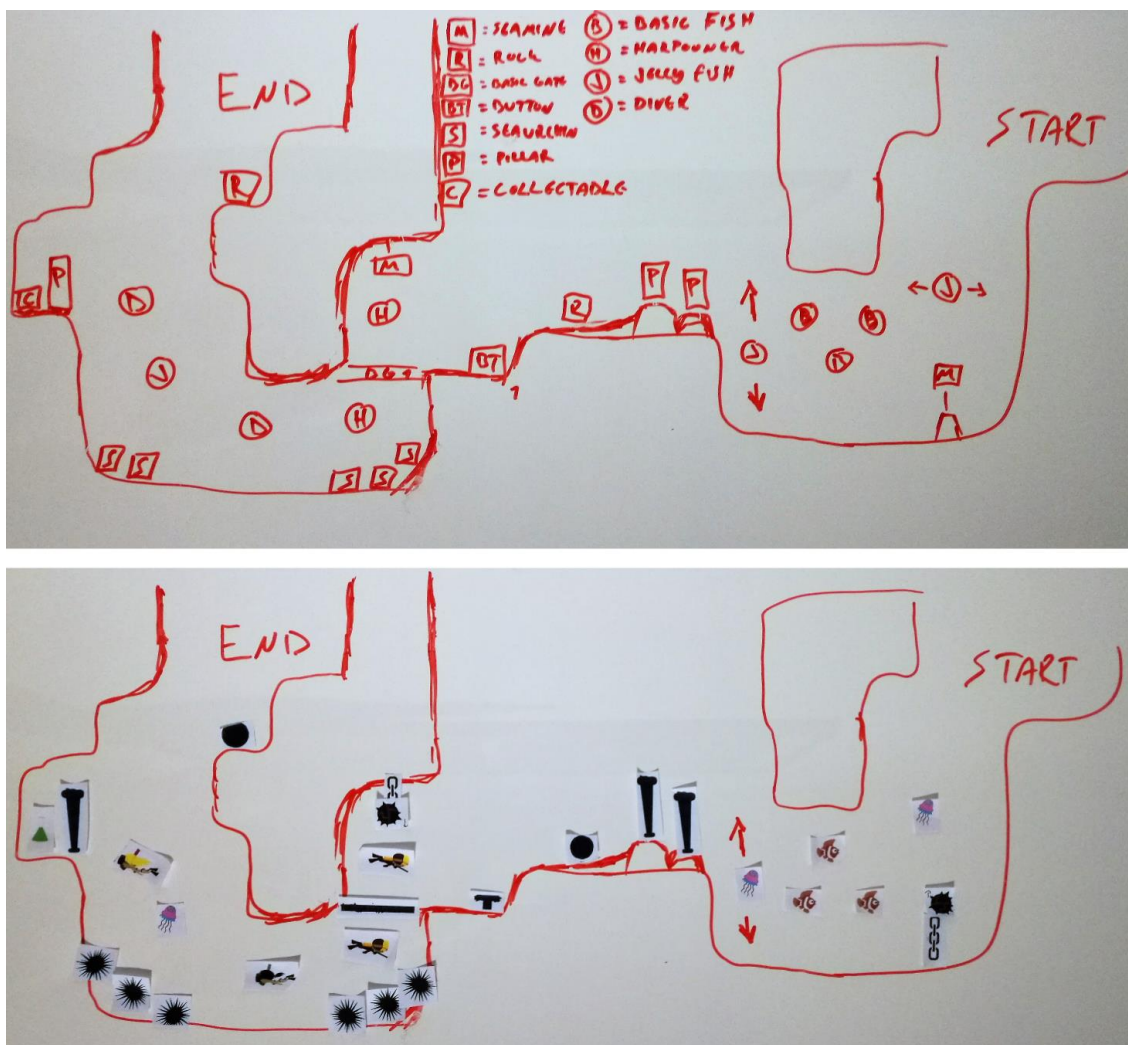
Niin kuin kuvasta 17 näkyy, symbolien käyttö toimi etenkin paperille piirrettyjen pohjapiirustusten kanssa, ja niiden avulla saatiin piirustukset pysymään selkeinä ja helposti luettavina.



Kuva 17. Hahmotusmenetelmä 1: Lyijykynällä tehty viimeistellympi pohjapiirustus. Pohjapiirustuksen vieressä nimettynä käytetyt symbolit.

Kun samat elementit laitettiin paikoilleen kuvilla, muodostui paperille piirrettyjen kanssa ongelmaksi mittakaava. Vaihtoehtona olisi ollut piirtää pohja isommalle paperille, jolloin piirustusten arkistointi ja muokkaaminen olisi ollut haastavampaa, tai pienentää kuvia, jolloin niistä olisi tullut vaikeammin tulkittavia.

Kuvan 18 avulla nähdään, kuinka leikatut kuvat toimivat erityisen hyvin, kun käytettiin taulua, jolle taso piirrettiin.



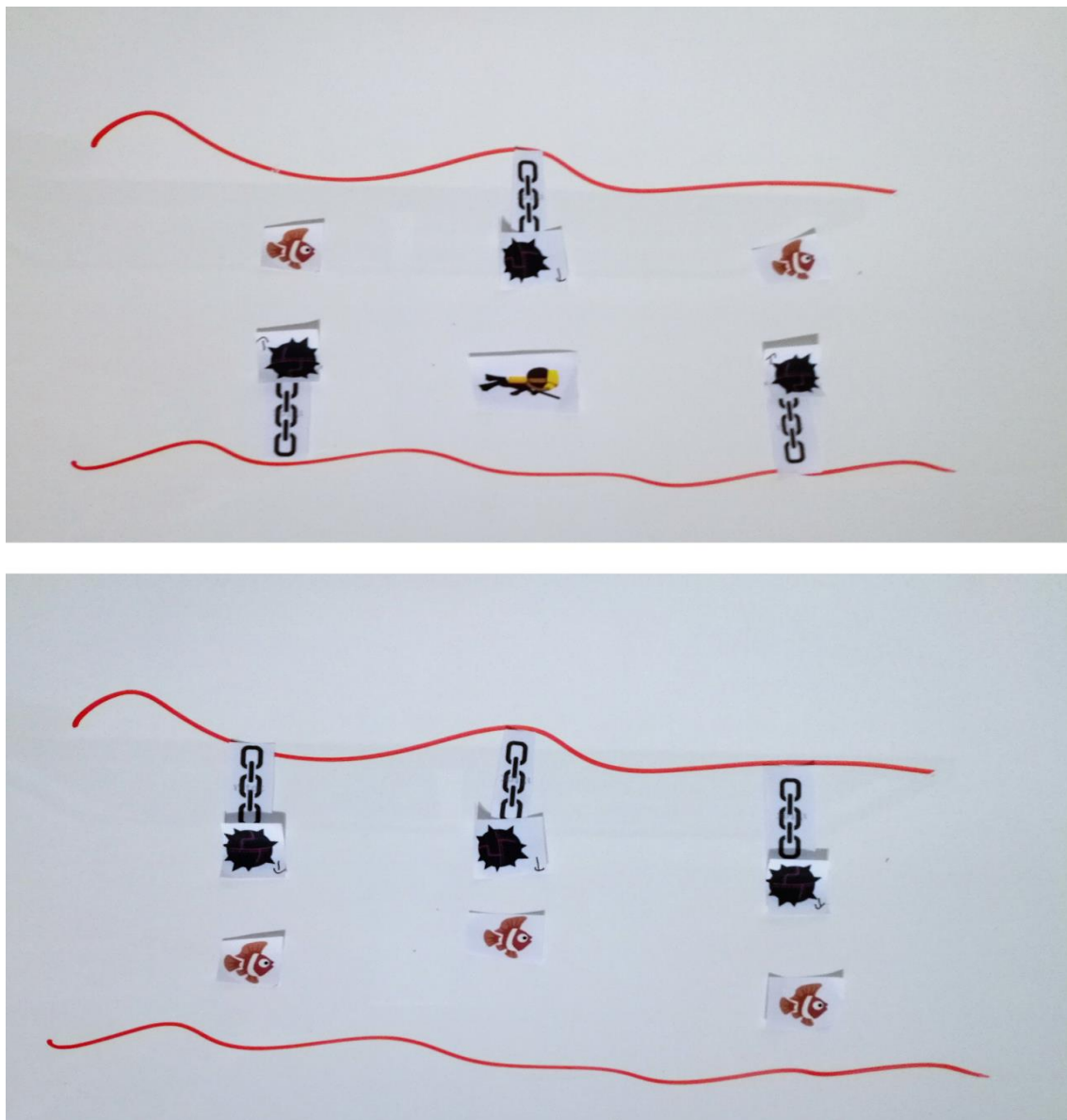
Kuva 18. Viimeistellymmän pohjapiirustuksen eroavaisuudet hahmotusmenetelmän 3 (ylempi taso) ja hahmotusmenetelmän 4 välillä.

Paperista leikatut kuvat elementeistä olivat myös huomattavasti nopeampi tapa kokeilla erilaista elementtien asettelua. Jokaisen tason kohdalla kokeiltiin useaa eri asettelua elementeillä, ja valmiiden pohjapiirustusten perusteella valittiin kaksi tasoa, jotka etenivät seuraavaan vaiheeseen eli kenttäeditorilla työstettäväksi. Nämä tasot valikoituivat, sillä ne sopivat parhaiten tasorakenteen osalta ja olivat myös hyvin lähellä toisiaan tasojen pituuden ja käytettyjen elementtien suhteen. Näin saatiin hyvin verrattua editorilla työstämisestä, mitä eroja saadaan aikaan eri menetelmillä.

Yksittäisten palojen suunnitteleminen oli huomattavasti vapaampaa, ja niitä luotiin neljäkymmentä. Aikaa palojen suunnitteluun meni noin kaksi tuntia. Suunnitteluprosessi erosi



erityisesti siinä, ettei näiden palojen kanssa tehty ensin yksinkertaisia pohjapiirustuksia ja lähdetty sen jälkeen viimeistelemään, vaan näin suunnitellut palat tehtiin alusta asti mittakaavaan piirtäen. Kuten kuvasta 19 voi nähdä, paloja sai myös testattua hyvin nopeasti eri elementtien kanssa, jolloin muokkaamiseen kulunut aika väheni selkeästi.



Kuva 19. Yksittäisten palojen piirtämistä ja useampien elementtien testaamista samalla pohjalla.

Tämä oli myös hyvä tapa tallentaa omia ideoita, joita ei lisännyt käytettäviin pohjapiirustuksiin. Osa paloista tuli myös tasoista, jotka eivät päätyneet kokonaisuudessaan kenttäeditorilla työstettäväksi, mutta niihin tehdyt osiot vaikuttivat muuten lupaavilta.

Kenttäeditorin käyttöön siirryttäessä oli etukäteen selvitetty tason kokoon liittyvät tiedot. Unityssa käytetään xyz-koordinaatistosta saatavien tietojen lisäksi omia Unity-yksiköitä, joilla voidaan tarkastella ruudukolle asetettujen elementtien kokoa ja niiden mittakaavaa. Unityn yksikkö on suhteessa metriin, jossa yksi Unity-yksikkö on yksi metri. Näitä yksiköitä käyttämällä määriteltiin keskimääräinen kulkuväylän leveys, jolloin kameran keskitäminen parveen, joka on keskellä kulkuväylää, näyttää pelaajalle sopivassa suhteessa tasoa ja parvea. Samoja yksiköitä hyödyntäen voitiin määrittää tason koko, jolloin oli huomattavasti helpompaa pitää eri tasot samassa skaalassa.

#### 4.4 Kentänluontimenetelmä 1: Kerrostus

Tasojen luominen aloitettiin käyttämällä kerrostusmenetelmää. Yksityiskohtaisten pohjapiirustusten takia nopeaan luonnokseen editorissa meni kymmenen minuuttia. Luonnoksella testattiin tason pituutta ja sitä, kuinka kulkuväyliä leveydet toimivat parven kanssa. Luonnosta muokattiin sen mukaan, miltä tasossa liikkuesssa eri kohdat vaikuttivat.

Vaikka tasojen pohjapiirustukset oli pyritty tekemään mahdollisimman hyvin oikeassa mittasuhteessa, kävi hyvin nopeasti selväksi, että jokainen kohta tasosta piti testata useaan otteeseen, jotta ne olisivat järkevissä mittasuhteissa muiden tason elementtien kanssa. Tähän oli kuitenkin varauduttu, sillä se oli yksi Ed Byrnen mainitsemista kohdista, joita tulee ottaa huomioon, kun tasoa tehdään. Kun luonnokseen oltiin tyytyväisiä, alettiin tasosta luoda tarkempaa ja viimeistellymmän näköistä. Tähän vaiheeseen kului aikaa 30 minuuttia, minkä aikana tasoa testattiin viisi kertaa. Jokaisen testauksen jälkeen jatkettiin tason muokkaamista, kunnes taso oli visuaalisesti silmää miellyttävä ja jokainen tason kohdista oli elementtien lisäystä vaille valmis. [13, s. 234.]

Prosessissa kului selkeästi eniten aikaa elementtien lisäysvaiheessa. Kaikki merkityt elementit asetettiin ensin karkeasti paikoilleen, minkä jälkeen jokainen kohta testattiin yksitellen useampaan otteeseen. Joidenkin kohtien osalla siihen suunnitellut elementit eivät tuntuneet toimivan, kuten oli aiemmin suunniteltu. Tämän vuoksi kahdessa kohdassa jouduttiin vaihtamaan elementit kokonaan. Tasoon myös lisättiin useampia elementtejä, kuin alkuperäiseen pohjapiirrokseen oli laitettu. Muutamat kohdat vaativat myös itse geometrisen tason muokkaamista, sillä kun niihin lisättiin kaikki elementit, oli osa kohdista hyvin ahtaita, jolloin niistä tuli pelaajalle liian vaikeita liikkua parven kanssa ja ne heikensivät pelin flow-tilaa. Elementtien lisäysvaiheeseen kului aikaa yhdeksänkymmentä

minuuttia, ja sinä aikana tasoa muokattiin kuusitoista kertaa. Tarkasteltaessa ensimmäistä vedosta tasosta, johon elementit oli lisätty, ja valmista tasoa oli ero huomattava. Siinä missä viimeistelyyn pohjapiirustukseen oli merkitty 21 elementtiä, tuli lopulliseen tasoon 38 elementtiä. Kuvasta 20 huomataan, kuinka alkuperäisen pohjapiirustuksen mukaan tehty taso eroaa viimeisestä versiosta ennen koristeellisten elementtien sijoittelua.



Kuva 20. Vertailussa alkuperäisen pohjapiirustuksen mukaan sijoitetut elementit ja lopullinen versio. Vasemmalla alkuperäinen ja oikealla lopullinen versio.

Viimeistelyvaiheessa parannettiin tason visuaalista ilmettä. Viimeistelyä varten käytettiin artistin luomia koristeellisia elementtejä ja vaihdettiin tason taustoja, jolloin saatiin lisättyä syvyyden tunnetta tasoon. Koristeellisten elementtien asettelulla saatiin erotettua maaston muotoja paremmin ja korostettua tahdottuja kohtia pelaajalle. Tämä voidaan havaita kuvasta 21. Koristeiden lisääminen tasoon vaikutti myös hyvin paljon muiden elementtien ja parven erottumiseen tasossa. Näin saatiin luotua kontrasti eri hahmojen ja tason välille, mutta pystyttiin kuitenkin säilyttämään pelin yhtenevä värimaailma.



Kuva 21. Tason viimeistelyä koristeellisilla elementeillä.

Viimeistelyvaiheeseen kului aikaa 30 minuuttia, ja sen aikana peliä testattiin viisi kertaa. Jokaisen testin jälkeen parannettiin elementtien sijoittelua ja tarkistettiin, että muiden elementtien toiminnallisuudet eivät häiriintyneet.

#### 4.5 Kentänluontimenetelmä 2: Ositus

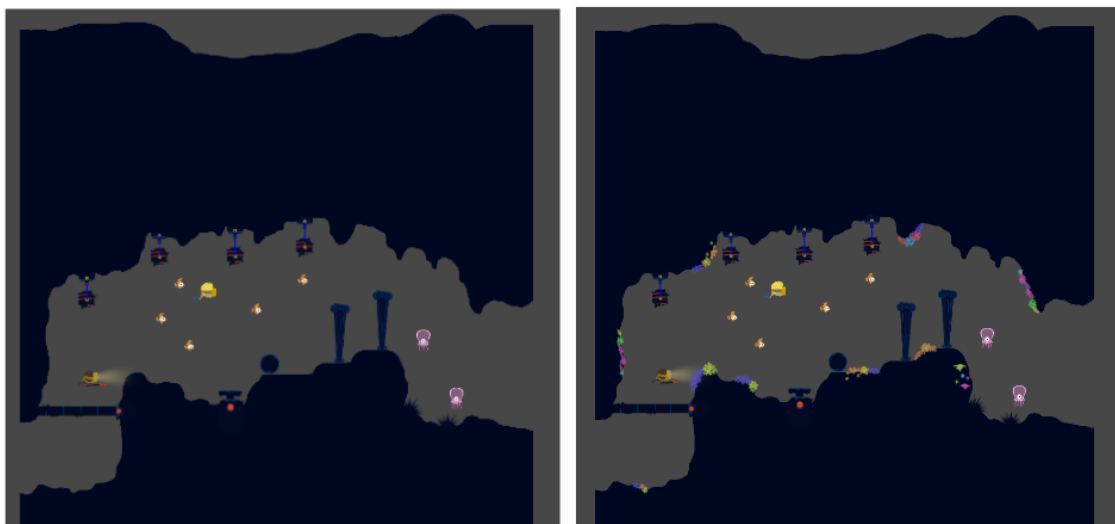
Kun ensimmäinen taso oli valmis, ryhdyttiin seuraavaa tasoa työstämään ositusmenetelmällä. Ennen editorilla työstämistä taso jaettiin kolmeen osaan, joista jokainen tehtiin vuoroittain valmiiksi ja tämän jälkeen yhdistettiin kokonaiseksi pelattavaksi tasoksi. Osittamisessa piti ottaa huomioon, miten taso jatkuu seuraavassa osassa ja kuinka nämä saadaan yhdistettyä toisiinsa saumattomasti. Itse osien yhdistäminen oli hyvin vaivastonta toimivien editorityökalujen ansiosta. Piirtotyökalulla tehdyt meshit yhdistettiin toisiinsa, minkä jälkeen taso luotiin kokonaisuudessaan. Kustakin osiosta luotiin jälleen

yksinkertainen pohja, jonka avulla katsottiin kulkuväylät ja mittasuhteet. Jokaiseen osioon kului aikaa 5 minuuttia, jona aikana testattiin kaksi kertaa osion toimivuus. Tämän jälkeen tasosta tehtiin tarkempi ja viimeistellympi versio. Viimeistellyn tason pohjan luomiseen kului kunkin osan kohdalla 15 minuuttia, jona aikana sen toimivuutta testattiin 5 kertaa. Kuvassa 22 nähdään, miten piirtotyökalulla on tehty mesh, josta luodaan pelissä käytettävän tason osa. Pohjan jälkeen lisättiin paikoilleen pelilliset elementit ja jatkettiin testausta. Samoin kuin ensimmäisen tason kohdalla, tuli jokaiseen osioon muutoksia testausten aikana ja kokonaiseen tasoon lisättyjen elementtien määrä nousi suuremmaksi kuin pohjapiirustukseen määritellyt.



Kuva 22. Tason mesh ja viimeistelty pohja ositusmenetelmää hyödyntäen.

Elementtien lisäämiseen ja testaamiseen käytettiin jokaisen osion kohdalla keskimäärin 20 minuuttia, ja niitä testattiin 8 kertaa. Osioden viimeistelyyn ja koristeluun kului kussakin osiossa keskimäärin 10 minuuttia, ja kutakin osiota testattiin viisi kertaa. Kun osiot oli saatu valmiiksi, ne piti vielä yhdistää kokonaiseksi tasoksi. Ennen tätä vaihetta oli ositusmenetelmällä luotuun tasoon kulunut aikaa 150 minuuttia, joka oli 10 minuuttia vähemmän kuin kerrostusmenetelmällä. Kuvassa 23 nähdään, kuinka osittamismenetelmällä on luotu osa tasoa ja miten koristeellisten elementtien lisääminen vaikuttaa tason ulkonäköön.



Kuva 23. Taso pelillisten elementtien sekä koristeellisten elementtien lisäyksen jälkeen.

Yhdistämisen aikana tasoa piti kuitenkin testata vielä useaan kertaan ja tarkistaa pelillisten elementtien ja koristeiden asettelu. Tasoa muokattiin 4 kertaa yhdistämisen aikana, jolloin korjattiin pohjan muotoja ja lisättiin koristeellisia elementtejä. Tähän kului aikaa 20 minuuttia. Kaiken kaikkiaan tasoon kului 170 minuuttia. Näin ollen kerrostusmenetelmä oli nopeampi tapa toteuttaa taso kuin osittamismenetelmä.

#### 4.6 Kentänluontimenetelmä 3: Valmiit palat

Kolmatta tasoa lähdettiin tekemään aikaisemmin luoduilla valmiilla paloilla. Taso oli tarkoitus luoda ilman erillistä kokonaisen tason suunnitteluprosessia ja kokeilla, miten tällä tavoin luotu taso saadaan toteutettua. Tässä vaiheessa pitää ottaa huomioon, että erillisten palojen suunnitteluun ja toteutukseen oli jo käytetty aikaa, ennen kuin varsinaista tasoa oli suunniteltu. Tasoa varten valittiin neljä etukäteen luotua palaa, jotka yhdistettiin yhdeksi kokonaiseksi tasoksi. Jokaisen palan väliin jätettiin riittävä välimatka, jotta piirityökalulla saatiin yhdistettyä meshit toisiinsa sujuvasti riippumatta palojen mallista. Tason luomiseksi paloilla kokeiltiin useita eri asetteluja, kunnes päästiin toimivan ja tyydyttävän näköiseen ratkaisuun. Paloissa selkeänä etuna oli se, kuinka helposti ja nopeasti visuaalinen toteutus tapahtui. Näin pystyttiin lyhyellä aikavälillä kokeilemaan montaa eri vaihtoehtoa. Palojen yhdistämisen jälkeen lisättiin pelillisiä elementtejä. Tämä vaihe vaati kuitenkin suhteellisen paljon aikaa, sillä vaikka paloihin oli sijoitettu valmiiksi

elementtejä, täytyi yhdistämisen jälkeen tarkastella tasoa kokonaisuutena, minkä vuoksi tasossa olevien pelillisten elementtien kokonaismäärä kasvoi merkittävästi. Paloissa valmiiksi olleita pelillisiä elementtejä oli 23, kun lopulliseen tasoon niitä tuli 42. Tason viimeistelyvaiheessa lisättiin koristeelliset elementit ja testattiin niiden vaikutus lopulliseen tasoon. Koska valmiissa paloissa ei ollut koristeellisia elementtejä aseteltuna ennakoon, meni tässä vaiheessa aikaa 30 minuuttia. Tason luonnin kokonaisajaksi muodostui 90 minuuttia, mutta tämä on osaltaan hieman harhaanjohtavaa sillä palojen suunnitteluun ja toteutukseen oli aiemmin käytetty useampia tunteja.

#### 4.7 Tasosuunnittelutyökalujen käyttö ja räätälöinti

Peliä varten luotiin yhdessä ohjelmoijan kanssa muutamia keskeisiä työkaluja tasojen tekemistä varten. Vaihtoehtona olisi ollut käyttää Unityn Asset Storesta löytyviä valmiiksi luotuja tasonluomistyökaluja, mutta kävi nopeasti ilmi, ettei niillä pysty toteuttamaan vaadittuja tasoja halutun mukaisesti ja niistä puuttui joitakin toiminnallisuuksia, joita haluttiin käyttää tason luonnissa.

Tasot haluttiin luoda käyttämällä piirtotyökalua, jolla käyttäjä voi piirtää määritellylle alueelle editorissa. Piirtotyökalulla piti pystyä luomaan ja poistamaan aiemmin tehtyä meshiä. Piirtämistä varten haluttiin luoda sivellintoiminnallisuus, joka loisi määritellyn kokoisen ympyrän jokaisella painalluksella. Tarkoituksena oli luoda nopeasti käytettävä työkalu, joka toimii suoraan editorissa, jolloin tasojen luominen ja muokkaaminen onnistuisi helposti ilman erillisiä ohjelmia. Tällä oli myös tarkoitus nopeuttaa testaamista, jolloin säästettäisiin aikaa, kun tasoa ei tarvitse siirtää ohjelmasta toiseen muokkausta varten.

Piirtotyökalu toimi ottamalla vastaan käyttäjän syöttämän hiiren painalluksen ja pitämällä yllä bittikarttaa näiden painallusten määrittelemistä paikoista. Kun käyttäjä painoi hiiren vasenta painiketta, työkalu maalasi määritellylle alueelle teksturoimattoman meshin. Tämä mesh oli visuaalinen ilmentymä ylläpidetystä bittikartasta, johon määriteltiin, mitkä bitit saavat arvon 1 eli aktiivinen tai 0 eli inaktiivinen. Siinä vaiheessa, kun piirretty taso oli valmis, luotiin viimeistelty taso hyödyntäen meshiä, jolloin saatiin halutuilla graafisilla ominaisuuksilla toteutettu taso, jossa oli myös colliderit eli fysiikoiden käyttämä törmäyspinta valmiina. Piirtotyökalulta vaadittiin seuraavia ominaisuuksia, jotta se olisi käyttökelpoinen tasojen toteuttamista varten:

- Työkalulla pitää pystyä piirtämään ja poistamaan meshiä.
- Piirtämiseen käytetyn siveltimen kokoa pitää voida muokata.
- Työkalun avulla tulee voida määrittää piirtoalueen koko, johon taso luodaan.
- Piirretyt meshit pitää voida tallentaa, ja niitä tulee voida muokata jälkikäteen.
- Erikseen luotuja meshejä tulee voida yhdistää.

Piirtotyökalun toteutuksessa oli alkuvaiheessa suuria ongelmia, sillä siinä luodut meshit olivat muodoiltaan hyvin karkeita, eikä tasaisen pyöreitä, kuten oli alun perin suunniteltu. Tämä saatiin ratkaistua piirretyn alueen optimoidulla interpoloinnilla, jossa bittikartassa olevien pisteiden paikkaa verrattiin siveltimen ympyrän kaareen. Suurimpia ongelmia oli kuitenkin erittäin epävakaa tallentaminen, sillä Unity kaatui usein siinä vaiheessa, kun meshiä alettiin viimeistellä tasoksi. Koska meshin tallentaminen oli määritelty tapahtuvaksi valmiin tason luonnin jälkeen, menetettiin myös kokonaan sillä kerralla toteutettu pohja. Ongelma paikannettiin valmiin tason viimeistelyssä tapahtuvaan meshin kolmioiden optimointiin, jolla on tarkoitus tehdä kentästä tarvittavan kevyt mobiilialustaa varten. Optimointiin tehtyjen muokkausten jälkeen editori ei kaatunut enää, ja varmuuden vuoksi piirretyn meshin tallentaminen siirrettiin tapahtuvaksi ennen varsinaisen tason luontia, jolloin vaikka ongelmia esiintyisi optimoinnissa, saataisiin pohja tallennettua.

Piirtotyökalu onnistuttiin toteuttamaan vaadittujen kriteereiden mukaisesti, ja sillä luotiin kaikki luvussa 4.3 mainitut tasot. Kuva 24 esittää, miten työkalulla on pystytty toteuttamaan haluttu tason mesh optimoitavaksi. Tasojen luonnin aikana piirtotyökaluun lisättiin muutamia ominaisuuksia, joilla pyrittiin parantamaan sen toiminallisuutta. Työkalulla haluttiin pystyä leikkaamaan ja liittämään valittu kohta piirretystä meshistä. Tällä tavoin pystyttäisiin pilkkomaan isoja kokonaisuuksia pienemmiksi paloiksi ja siirtämään osia meshien välillä. Leikkaus tehtiin aktivoimalla työkalusta alueenmaalausominaisuus, jolloin painamalla hiiren vasenta tai oikeaa painiketta voi maalata halutun kokoisen alueen. Kun haluttu alue oli maalattu, se voitiin leikata omaksi erilliseksi objektiksi, jota voitiin liikuttaa haluttuun kohtaan määritellyllä piirtoalueella ja liittää.





Kuva 24. Piirtotyökalulla luotu mesh ennen optimointia.

Unityn tarjoamista työkaluista hyödynnettiin erityisesti sen mahdollisuutta luoda prefabeja eli objekteista ennakkoon määriteltyjä ilmentymiä, joille oli määritelty omat toiminnallisuudet ja parametrit. Tasoissa käytettyjen pelillisten elementtien luominen prefabeina nopeutti prosessia elementtienlisäysvaiheessa, jolloin jokaista elementtiä ei tarvinnut erikseen luoda. Tämä oli myös osa optimointiin liittyviä aiheita, sillä käytettäessä samaa prefabia ei tarvitse luoda monta erilaista objekta vaan voidaan käyttää ilmentymää yhdestä jo määritellystä.

## 5 Tasosuunnitteluprosessin tarkastelu

### 5.1 Tulokset ja kokonaisuus

Kullakin luvussa esitellyllä ja käytetyllä menetelmällä saatiin aikaan hyvin toimivat tasot. Tasojen luonnissa saatiin tuloksissa aikaan riittävä määrä eroavaisuuksia, jotta voidaan tarkastella niiden vaikutuksia pidempiaikaisen projektin tasosuunnitteluun. Menetelmien väliset erot korostuivat erityisesti siinä vaiheessa, kun tasoja alettiin luoda editorilla. Tasojen luonnin alussa tapahtunut suunnitteluvaihe oli identtinen kahdelle ensimmäiselle kentänluontimenetelmälle. Kolmantena kentänluontimenetelmänä esitetty paloista koostuva tason luominen eroaa jo lähtökohtaisesti suunnittelutyön alustuksessa, jolloin siitä saatava vertailukelpoinen data täytyy käydä läpi hieman eri tavalla eikä se ole

yksittäisenä tapahtumana suoraan verrattavissa kahteen ensimmäiseen. Kun tarkastellaan laajemmin tasosuunnittelua ja siihen käytettyjä menetelmiä, pystytään paremmin huomioimaan käytetty aika ja muut resurssit sekä niillä aikaansaadut tulokset, jolloin myös kolmantena esitetty kentänluontimenetelmä on paremmin vertailtavissa.

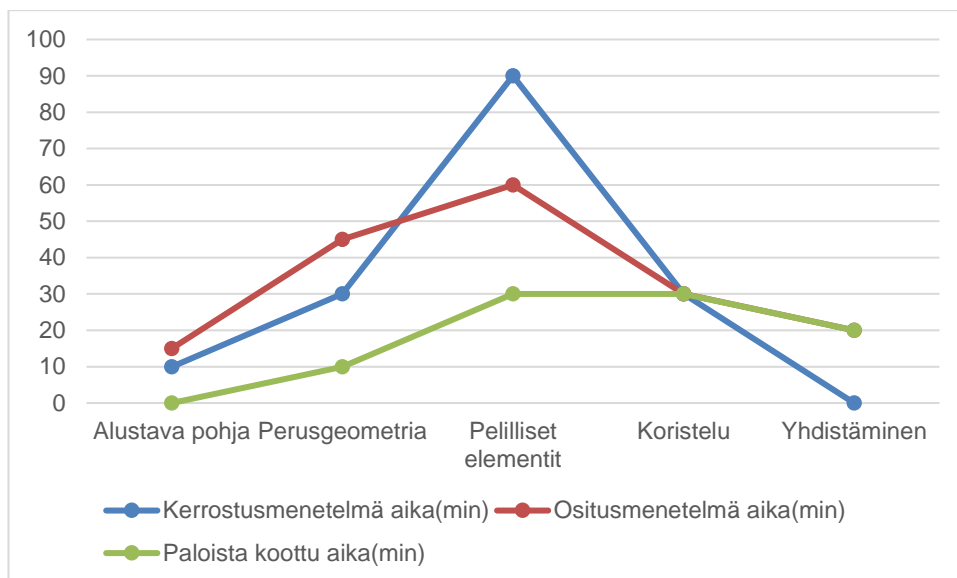
Alustavassa suunnittelussa muodostui tärkeäksi tarvittavien tietojen kerääminen ja niiden organisointi. Suunnittelutyötä oli huomattavasti helpompi lähestyä sen jälkeen, kun oli selvittänyt millaisilla elementeillä, mekaniikoilla ja tasorakenteella ollaan lähdössä työstämään tasoja. Piirrettäessä kokonaisiasa tasoja saatiin eri tavoilla aikaan suhteellisen vähän eroja. Näin saatujen kokemusten perusteella vaikuttaisi olevan tärkeintä se, millä tavalla itse saa paremman ja selkeämmän lopputuloksen aikaan. Molemmilla tavoilla, eli paperille piirrettäessä ja taulua käyttäessä, oli positiiviset ja negatiiviset puolensa. Paperin käyttö oli parempi jatkokäsittelyä ja omien symboleiden käyttöä varten, kun taas käytettäessä leikattuja kuvia elementeistä ja halutaan isompi ja selkeämpi kuva piirretystä tasosta oli taulun käyttäminen huomattavasti parempi vaihtoehto. Tasojen piirtämiseen kulunut aika oli miltei sama molemmilla tyyleillä ja riippui enemmän tason pohjasta kuin käytetystä tavasta. Yksittäisiä paloja ja kohtauksia tehdessä pätivät myös samat kohdat, kuin kokonaisiasa tasoja piirrettäessä. Eroja kokonasiin tasoihin muodostui nopeudessa, sillä yksittäisiin paloihin kului keskimäärin kaksi minuuttia, siinä missä kokonaisen tason suunnitteluun ensimmäisestä vedoksesta viimeistelyyn pohjapiirustukseen, jossa oli elementit aseteltuina, meni viisitoista minuuttia.

Editorilla työskenneltäessä saatiin aikaan eriäviä tuloksia, jotka eivät kahden ensimmäisen menetelmän osalta olleet kuitenkaan merkittäviä ajallisesti, kun taas testikertojen määrissä tapahtui yli kaksinkertainen nousu. Kuvasta 25 voidaan tarkastella, kuinka kerrostusmenetelmää käytettäessä yksittäisen tason luontiin kului aikaa 160 minuuttia ja tasoa testattiin kaikkiaan 28 kertaa, kun taas ositusmenetelmää hyödynnettäessä aikaa tason luomiseen kului 170 minuuttia, mutta tasoa testattiin 64 kertaa. Tämä selittyy yksinkertaisesti sillä, että jokainen pala piti testata erikseen ja tasoa testattiin vielä yhdistämisen aikana. Ositusmenetelmässä ei kuitenkaan testausmäärästä huolimatta kestänyt kuin 10 minuuttia kauemmin kuin kerrostusmenetelmää käytettäessä. Syy tähän on, että pienempien osioiden testaaminen vie myös vähemmän aikaa, ja tämä onkin positiivinen piirre, sillä tasoja pitäisi testata mahdollisimman paljon ja usein.

Kerrostusmenetelmä	Aika (min)	Testikerrat
Alustava pohja	10	2
Perusgeometria	30	5
Pelilliset elementit	90	16
Koristelu	30	5
Yhdistäminen	0	0
<b>Yhteensä</b>	<b>160</b>	<b>28</b>
Ositusmenetelmä	Aika (min)	Testikerrat
Alustava pohja	15	2
Perusgeometria	45	5
Pelilliset elementit	60	8
Koristelu	30	5
Yhdistäminen	20	4
<b>Yhteensä</b>	<b>170</b>	<b>64</b>
Paloista koottu	Aika (min)	Testikerrat
Alustava pohja	0	
Perusgeometria	10	5
Pelilliset elementit	30	8
Koristelu	30	5
Yhdistäminen	20	10
<b>Yhteensä</b>	<b>90</b>	<b>28</b>

Kuva 25. Tasojen luontiin editorissa käytetyt ajat ja testikertojen määrät.

Kun verrataan suoraan taulukossa olevaa dataa keskenään ja tarkastellaan kuvassa 26 olevaa viivakaaviota, voidaan todeta, että paloista kokoamalla saadaan tuotettua tasoja yli 1,5 kertaa nopeammin ja testikertoja kertyi saman verran kuin kerrostusmenetelmää käytettäessä. Tämä on, kuten aiemmin mainittu, hieman harhaanjohtavaa, kun käsitellään yksittäistä tasoa puhtaasti siihen käytetyn ajan perusteella. Mikäli tason luontiin lasketaan mukaan kaikkien 40 palan tekeminen, nousee tällä tavoin tehtyyn tasoon käytetty aika. Jokaisen palan luomiseen editorilla kului keskimäärin 15 minuuttia. Näin ollen kokonaisuudessaan paloihin käytettiin aikaa 600 minuuttia. Jos tämän laskee mukaan yhden tason toteuttamiseen käytettyyn aikaan, siitä tulee kohtuuttoman pitkä eikä se vastaa tarkoitustaan.



Kuva 26. Eri menetelmillä käytetty aika osioittain.

Jotta tuloksia voidaan verrata keskenään, pitää ottaa huomioon koko pelin tasojen määrä. Mikäli kaikki 80 tasoa tehtäisiin kerrostusmenetelmällä, kuluisi siihen tällä nopeudella 213 tuntia, kun taas ositusmenetelmällä menisi aikaa 226 tuntia. Mikäli taas luotaisiin paloista koottuja tasoja, kuluisi tasojen tekemiseen 130 tuntia, johon on laskettu mukaan 10 tuntia, joka on palojen luontiin käytetty aika. Jos oletusarvoisesti olisi käytettävissä 40 palaa ja jokaisen tason luontiin käytettäisiin neljää eri palaa, olisi kyseisellä määrällä mahdollista tehdä 91,390 erilaista yhdistelmää. Tämä saadaan laskemalla kaavalla

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Näissä pitää kuitenkin huomioida, että samojen palojen toisto voi näkyä pelaajalle, ja samoin etteivät kaikki palat sovi yhteen esimerkiksi tasorakenteen osalta. Tasojen osalta havaittiin myös, että kerrostus- ja ositusmenetelmällä tulee laadullisesti paremman oloisia tasoja. Tämä johtuu selvästi tason pohjapiirustuksen luontiin ja näin ollen suunnitteluun käytetystä ajasta.

## 5.2 Esitettyjen menetelmien ja työkalujen hyödyntäminen jatkossa

Koska peli on tasojen osalta vielä alkuvaiheessa, on mahdollista jatkaa kokeilua suunnittelumenetelmien osalta. Jokaisessa tavassa oli omat hyvät puolensa, ja jatkossa todennäköisesti hyödynnetään jokaista tässä insinööriyössä käytettyä menetelmää. Näin

saadaan pidettyä työ vaihtelevana ja mielenkiintoisena, mikä on tärkeää, kun tehdään luovaa työtä. Menetelmiä tullaan kuitenkin käyttämään tukien toisiaan, ja esimerkiksi ositusmenetelmää käytettäessä voidaan samalla luoda paloja, joita voidaan jatkossa hyödyntää muihinkin tasoihin. Samalla tavalla pyritään toimimaan kerrosmenetelmällä luotujen tasojen kanssa, sillä niissä olevat meshit voidaan pilkkoa sopiviin osiin hyödyntäen piirtotyökalua.

Editorissa olevia työkaluja pitää kehittää vielä jatkossa, jotta ne ovat täysin luotettavia eikä niiden käyttämisestä koidu turhaa työtä. Tällä hetkellä niiden toiminnallisuudet ovat kuitenkin riittävän stabiilit ja niillä saadaan tehtyä tarkoitusta vastaavaa työtä.

## 6 Yhteenveto

Insinööriyön tavoitteena oli perehtyä erilaisiin tasosuunnittelumenetelmiin ja työkaluihin ja luoda erityisesti mobiilipelien tasosuunnittelulle pohjaa, jolla voidaan jatkossa rakentaa toimivia kokonaisuuksia. Kun tarkastellaan työn aikana tehtyjä suunnittelutöitä ja tasoja, voidaan todeta, että tavoitteessa onnistuttiin. Vaikka työ keskittyi mobiilipeleihin, on tasosuunnittelussa paljon elementtejä, jotka pätevät riippumatta pelin alustasta. Tässä työssä onnistuttiin mielestäni osoittamaan, miten universaalit suunnittelumenetelmät ja työkalut toimivat.

Työssä luotujen ja käytettyjen työkalujen kehittämisessä on parannettavaa erityisesti niiden suunnittelussa. Luodut työkalut saatiin kuitenkin toimiviksi, ja niiden avulla voidaan toteuttaa tarvittavat tasot tämän insinööriyön kohteena olleeseen peliin.

Työstä saadun datan avulla pystyttiin tarvittavan tarkasti vertaamaan useamman eri menetelmän vaikutusta suunnitteluprosessiin ja sen pituuteen. Saadun datan pohjalta voidaan tulevaisuudessa miettiä, kuinka tason suunnitteluprosessissa voidaan hyödyntää eri menetelmiä ja kehittää jatkossa prosessia tehokkaammaksi. Viimeiseksi käytetystä menetelmästä on kuitenkin huomioitava, että sen kanssa oli haastavaa tehdä kokonainen taso, mikä johtui kokonaisen tason piirustuksen puuttumisesta. Tähän voisi jatkossa kehitellä muutosta, sillä tason luomiseen muuten käytetty aika oli huomattavasti lyhyempi.

Insinööritö opetti erityisesti miten tasosuunnitteluprosessia lähdetään rakentamaan ja kuinka noudattamalla tarkkaan laadittua suunnitelmaa saadaan aikaan toimivia kokonaisuuksia, joita pystyttiin luomaan annettujen vaatimusten mukaisesti. Työssä opittiin myös Unityn editorissa käytettävien työkalujen käsittelyä ja toimintaa. Insinööritössä esillä olleet tasot jatkavat seuraavaan vaiheeseen pelinkehitysprosessissa.

## Lähteet

- 1 Hiltunen, Koopee; Latva, Suvi; Kaleva, J-P. 2017. The Game Industry of Finland. Verkkoaineisto. Neogames. <[http://www.neogames.fi/wp-content/uploads/2017/04/Finnish-Game-Industry-Report-2016\\_web\\_070529.pdf](http://www.neogames.fi/wp-content/uploads/2017/04/Finnish-Game-Industry-Report-2016_web_070529.pdf)>. Luettu 1.2.2018.
- 2 McDonald, Emma. 2017. The Global Games Market Will reach \$108.9 Billion in 2017 With Mobile Taking 42%. Verkkoaineisto. Newzoo. <<https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/>>. Luettu 2.2.2018.
- 3 Lappalainen, Elina. 2016. Suomen peliala kasvaa komeasti: Euroopassa kolmen kärjessä. Verkkoaineisto. TiVi. <[https://www.tivi.fi/Kaikki\\_uutiset/suomen-peliala-kasvaa-komeasti-euroopassa-kolmen-karjessa-6426167](https://www.tivi.fi/Kaikki_uutiset/suomen-peliala-kasvaa-komeasti-euroopassa-kolmen-karjessa-6426167)>. Luettu 2.2.2018.
- 4 Game production pipeline. 2017. Verkkoaineisto. The Knights of Unity. <<http://blog.theknightsofunity.com/game-production-pipeline/>>. Luettu 6.2.2018.
- 5 The game production pipeline: Concept to completion. 2006. Verkkoaineisto. IGN. <<http://www.ign.com/articles/2006/03/16/the-game-production-pipeline-concept-to-completion>>. Luettu 3.2.2018
- 6 Ryan, Tim. 1999. The Anatomy of a Design Document, Part 1. Verkkoaineisto. Slideshare <<https://www.slideshare.net/shenerd/guide-to-creation-of-game-concept-document>>. Luettu 6.2.2018
- 7 Brathwaite, Brenda. 2008. Types of Game Designers. Verkkoaineisto. GameCareerGuide. <[https://www.gamecareerguide.com/features/483/types\\_of\\_game\\_.php](https://www.gamecareerguide.com/features/483/types_of_game_.php)>. Luettu 6.2.2018.
- 8 Some questions about OW development. 2016. Verkkoaineisto. Battlenet forum. <<https://us.battle.net/forums/en/overwatch/topic/20748786270#post-12>>. Luettu 6.2.2018
- 9 Munford, Monty. 2010. Angry Birds: 10 things you probably didn't know. Verkkoaineisto. Telegraph. <<https://www.telegraph.co.uk/technology/video-games/8058622/Angry-Birds-10-things-you-probably-didnt-know.html>>. Luettu 12.4.2018
- 10 Wade, Nicholas J; Swanston, Michael T. 2013. Visual Perception, An Introduction, 3<sup>rd</sup> Edition. Psychology Press.
- 11 Biol, J. 2008. Music, memory and emotion. Verkkoaineisto. NCBI. <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2776393/>>. Luettu 10.2.2018.

- 12 Scolastici, Claudio & Nolte, David. 2013. Mobile Game Design Essentials. Birmingham: Packt Publishing.
- 13 Byrne, Ed. 2005. Game Level Design. Charles River Media, Inc.
- 14 Hwong, Connie. 2016. The Average Mobile Game Day. Verkkoaineisto. Verto Analytics. <<http://www.vertoanalytics.com/average-mobile-game-day/>>. Luettu 10.2.2018.
- 15 Bradshaw, Hazel. 2016. Gameplay Flow -Designing for Player Immersion. Verkkoaineisto. Gamasutra. <[https://www.gamasutra.com/blogs/Hazel-Bradshaw/20160525/273468/Gameplay\\_Flow\\_\\_Designing\\_for\\_Player\\_Immersion.php](https://www.gamasutra.com/blogs/Hazel-Bradshaw/20160525/273468/Gameplay_Flow__Designing_for_Player_Immersion.php)>. Luettu 12.2.2018.
- 16 Lopez, Mike. 2006. Gameplay Design Fundamentals: Gameplay Progression. Verkkoaineisto. Gamasutra. <[https://www.gamasutra.com/view/feature/130188/gameplay\\_design\\_fundamentals\\_.php?page=5](https://www.gamasutra.com/view/feature/130188/gameplay_design_fundamentals_.php?page=5)>. Luettu 12.2.2018.
- 17 Jonkers, Diorgo. 2011. How to design levels for a platformer. Verkkoaineisto. Dev.Mag. <<http://devmag.org.za/2011/07/04/how-to-design-levels-for-a-platformer/>>. Luettu 12.2.2018.
- 18 Galuzin, Alex. 2011. Ultimate Level Design Guide. WorldofLevelDesign.com
- 19 Unity Manual. Importing from the Asset Store. Verkkoaineisto. Unity. <<https://docs.unity3d.com/Manual/AssetStore.html>>. Luettu 4.2.2018.
- 20 Unreal Editor Manual. Level Editor. Verkkoaineisto. Epic Games. <<https://docs.unrealengine.com/en-US/engine/ui/leveleditor>>. Luettu 4.2.2018.
- 21 Galuzin, Alex. 2012. What Level Editor and Game Engine Should You Use – (How to Choose). Verkkoaineisto. WorldOfLevelDesign. <[http://www.worldofleveldesign.com/categories/level\\_design\\_tutorials/what-level-editor-game-engine-should-you-use-how-to-choose.php](http://www.worldofleveldesign.com/categories/level_design_tutorials/what-level-editor-game-engine-should-you-use-how-to-choose.php)>. Luettu 4.2.2018.
- 22 Rucker, Michael. What is Fun Anyway? Verkkoaineisto. Michael Rucker <<https://michaelrucker.com/having-fun/what-is-fun/>>. Luettu 12.2.2018.
- 23 Best Fiends – Puzzle Adventure. 2018. Verkkoaineisto. Google Play. <<https://play.google.com/store/apps/details?id=com.Seriously.BestFiends>>. Luettu 16.2.2018
- 24 Angry Birds Wiki. 2017. Poached Eggs 1-21. Verkkoaineisto. Fandom. <[http://angrybirds.wikia.com/wiki/Poached\\_Eggs\\_1-21](http://angrybirds.wikia.com/wiki/Poached_Eggs_1-21)>. Luettu 12.4.2018.



- 25 Screenshot: Reinhardt. 2018. Verkkoaineisto. Blizzard Entertainment.  
<<https://playoverwatch.com/en-us/media/reinhardt-screenshot-006>>. Luettu  
12.4.2018.
- 26 PlayerUnknown's Battlegrounds Mobile. 2018. Verkkoaineisto. Google Play.  
<<https://play.google.com/store/apps/details?id=com.tencent.ig>>. Luettu  
12.4.2018.
- 27 Valve Hammer Editor. 2017. Verkkoaineisto. Valve Developer Community.  
<[https://developer.valvesoftware.com/wiki/Valve\\_Hammer\\_Editor](https://developer.valvesoftware.com/wiki/Valve_Hammer_Editor)>. Luettu  
12.4.2018.

**Viimeistellyt tasot käytettyjen menetelmien mukaisessa järjestyksessä**



